TECHNISCHE UNIVERSITÄT DARMSTADT

Technische Universität Darmstadt
Department of Computer Science
Chair of Applied Cryptography

Master Thesis

Trustless incoercible sale of physical goods over a blockchain

Nikolaos Stivaktakis

October 11, 2022

Supervisors: Prof. Sebastian Faust, Ph.D.
Orfeas Stefanos Thyfronitis Litos, Ph.D.

Abstract

As digitalization is becoming an essential part within our society, cryptocurrencies are rapidly gaining acceptance as a valid payment method for physical goods. Usually the sales of a physical good involves a trusted third party which ensures that the Seller gets payed and the Buyer receives the physical good. As the third party can misbehave or no trusted third party is available at all, there is a rising need for solutions that do not rely on a third party. This led to us to our research topic:

Investigate the existence of a protocol that enables a trustless and incoercible sale of physical goods over a blockchain.

We formalize the problem using a game-theoretical approach. To find a solution without a trusted third party we iteratively propose solutions and assess if this solution satisfies the success criteria. As this approach did not lead us to a viable solution we later proof that there exists no such a two-party solution to the formalized problem. Afterwards we modify the problem by relaxing the preconditions and allowing a third party to be apart of the solution. Finally we successfully present a solution to the modified problem.

!!! Prüfen Sie, dass der folgende Text aktuell ist (entsprechend der formalen Regeln des Studienbüros) !!!
!!! Check that this text is up to date (according to formal rules of the examination office) !!!

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 APB TU Darmstadt

Hiermit versichere ich, **Nikolaos Stivaktakis**, die vorliegende Master-Thesis / Bachelor-Thesis gemäß § 22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.
Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.
Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

English translation for information purposes only: Thesis Statement pursuant to § 22 paragraph 7 of APB TU Darmstadt

I herewith formally declare that I, **Nikolaos Stivaktakis**, have written the submitted thesis independently pursuant to § 22 paragraph 7 of APB TU Darmstadt. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form.
I am aware, that in case of an attempt at deception based on plagiarism (§38 Abs. 2 APB), the thesis would be graded with 5,0 and counted as one failed examination attempt. The thesis may only be repeated once.
For a thesis of the Department of Architecture, the submitted electronic version corresponds to the presented model and the submitted architectural plans.

Datum / Date

Unterschrift / Signature

List of Figures

2.1 Example Tree 6

4.1 Protocol: No collateral, Buyer sends first 12
4.2 Protocol: No collateral, Seller sends first 12
4.3 Protocol: No collateral, both players send at simultaneously 12
4.4 No collateral, Buyer acts first 13
4.5 No collateral, Seller acts first 13
4.6 Buyer collateral protocol 14
4.7 Only the Buyer deposits collateral 15
4.8 Seller collateral protocol 16
4.9 Only the Seller deposits collateral 17
4.10 Seller collateral protocol: Both parties report 18
4.11 Seller collateral protocol: Both parties report 19
4.12 Only the Seller deposits collateral: Buyer reports first 20
4.13 Only the Seller deposits collateral: Both players report simultaneously 21
4.14 Two side collateral protocol 23
4.15 Two side collateral protocol: Seller reports first 24
4.16 Two Side collateral protocol: Seller reports first 24
4.17 Two side collateral protocol: Buyer reports first 25
4.18 Two side collateral protocol: Buyer reports first 26
4.19 Two side collateral protocol: Simultaneous report 27
4.20 Two side collateral protocol: Both parties report simultaneously 27
4.21 General Code for the ideal functionality 29
4.22 Example Code for an implementation of the ideal Smart Contract 30
4.23 Payout with an ideal Smart Contract 30
4.24 Signature of a function realizable in the real world 31

5.1 Two Side Collateral with truthful Buyer 33
5.2 Mediated protocol description 35
5.3 Mediated Protocol 36
5.4 Mediated protocol: honest Buyer behaviour 37
5.5 Mediated protocol: honest Seller behaviour 37
5.6 Two Side Collateral with ideal Mediator 37
5.7 Two side collateral protocol: Friend of Buyer reports 39
5.8 Two side collateral protocol: Postal Office as Mediator 40
5.9 Two side collateral protocol: Machine as Mediator 41

List of Tables

2.1 Example Table 7

4.1 No collateral: Both parties send simultaneously 14
4.2 The corresponding payout table of figure 4.11 19
4.3 The corresponding payout table of figure 4.12 20
4.4 Subgame T1: Both players report simultaneously 21
4.5 Subgame T2: Both players report simultaneously 22
4.6 The corresponding payout table of figure 4.16 24
4.7 The corresponding payout table of figure 4.18 26
4.8 Subgame T1: Both parties report simultaneously 27
4.9 Subgame T2: Both parties report simultaneously 27
4.10 The corresponding payout table of figure 4.23 31

5.1 The corresponding payout table of figure 5.1 34

Contents

List of Figures iv

List of Tables v

1 Introduction 1
- 1.1 Research Topic . . . 2
- 1.2 Methodology . . . 2
- 1.3 Structure of this thesis . . . 3
- 1.4 Related work . . . 3

2 Preliminaries 5
- 2.1 Game Theory . . . 5
  - 2.1.1 Games . . . 5
  - 2.1.2 Strategies . . . 5
  - 2.1.3 Nash Equilibrium . . . 6
  - 2.1.4 Two-party alternating turn-based games . . . 6
  - 2.1.5 Two-party simultaneous games . . . 7
- 2.2 Commitment schemes . . . 7
- 2.3 Blockchains . . . 8
- 2.4 Smart Contracts . . . 8

3 Formalization 10
- 3.1 Entities . . . 10
- 3.2 The Game . . . 10

4 Candidate Two-Party Protocols 12
- 4.1 First try: no collateral . . . 12
  - 4.1.1 Possible strategies . . . 12
  - 4.1.2 Game: The Buyer acts first . . . 13
  - 4.1.3 Game: The Seller acts first . . . 13
  - 4.1.4 Game: Both parties act simultaneously . . . 14
- 4.2 Second Try: Buyer Collateral . . . 14
  - 4.2.1 Possible Strategies . . . 15
  - 4.2.2 Game . . . 15
- 4.3 Third Try: Seller Collateral . . . 16
  - 4.3.1 Possible Strategies . . . 16
  - 4.3.2 Game . . . 16
- 4.4 Fourth try: Seller collateral, both parties report . . . 18
  - 4.4.1 Possible Strategies . . . 18
  - 4.4.2 Game: The Seller reports first . . . 18
  - 4.4.3 Game: The Buyer reports first . . . 20
  - 4.4.4 Game: Both parties report simultaneously . . . 21
- 4.5 Fifth Try: Two side collateral . . . 22
  - 4.5.1 Possible strategies . . . 22
  - 4.5.2 Game: The Seller reports first . . . 24
  - 4.5.3 Game: The Buyer reports first . . . 25

4.5.4 Game: Simultaneous report . . . 27
4.6 An extra input is needed . . . 28
4.6.1 Ideal Functionality . . . 28
4.6.2 Ideal Smart Contract solves the proposed problem . . . 29
4.6.3 Real World Functionality . . . 31

5 A redesigned game **33**
5.1 Two Side Collateral with truthful Buyer . . . 33
5.2 Formal definition of the new Entity . . . 35
5.3 Mediated protocol . . . 35
5.3.1 Honest behaviour . . . 37
5.3.2 Possible Strategies . . . 37
5.4 Possible realizations of Mediator . . . 38
5.4.1 Friend of Buyer reports . . . 38
5.4.2 Postal Office reports . . . 40
5.4.3 Machine reports . . . 41
5.5 Is the proposed Mediator a trusted third party? . . . 42

6 Conclusion **43**
6.1 Topics for future work . . . 43

1 Introduction

Technological advancement is ongoing. In the last years there has been a big rise of the digital world. Traditional processes in many aspects of human life have been brought from the real, physical world to the digital world. This transformation has been accelerated through the Covid-19 pandemic. One traditional aspect of life that is being digitized in particular the last years is money. While banks allow digital payments for quite some years now, until the recent time there was no digital counterpart to cash.

While there were approaches for digital cash (e.g.[Cha83]), none of the approaches prevailed and gained acceptance as a valid option to transfer monetary value. That is until the Bitcoin whitepaper [Nak08] was released and created the wide field of cryptocurrencies. As the title "Bitcoin: A Peer-to-Peer Electronic Cash System" suggests, Bitcoin aimed to be a decentralized implementation of digital cash. With such a system one should be able to make fast, direct and anonymous payments to any other person that is also using the system. Each of these payments, called transactions, are written on the **blockchain**, a distributed database that acts as a public ledger. After Bitcoin many other cryptocurrencies were introduced, most notably Ethereum [But14]. Besides the transfer of value, Ethereum also aimed to run code in a distributed manner over the blockchain. These automated programs, called Smart Contracts, can be used to automate the execution of an agreement between multiple parties.

As cryptocurrencies are gaining acceptance, they become a valid payment option for sales of digital and physical goods. Two parties are involved in this transaction: A Seller with the obligation to send the good and a Buyer with the obligation to pay the agreed amount. The remote and anonymous nature of blockchains complicates these sales, because the trading parties cannot always trust each other. An adherent problem arises: The party that first fulfills its obligation, risks getting scammed by the opposing party not fulfilling its obligation. This problem is commonly solved using a trusted third party, an escrow [Gol+17]. This escrow first collects the payment and the good and then forward them to their recipients.

While this solution is working, there are several reasons why a protocol without a trusted third party is of interest. There is always a risk that the third party misbehaves and keeps both the good and the price. In addition to that there are scenarios, where there does not exist a third party that is trusted by both the Buyer and the Seller. In these scenarios the two parties can not trade, even if a trade would benefit both parties. Therefore there is a need for a protocol without a trusted third party. We will call these protocols **trustless**.

If the traded good is digital, such a trustless protocol can be achieved by building a Smart Contract that atomically exchanges the digital good for its price [AK19]. Creating a Smart Contract that atomically exchanges a **physical good** for its price is much more complicated, because physical goods and their delivery are unknown to the blockchain. In contrast to digital goods, the delivery fulfillment of the correct physical goods is commonly not written on the blockchain. Therefore a Smart Contract can not access this information and does not know if the Seller did fulfill his obligation.

The first approaches to address the sale of physical goods (e.g. [Zim]) introduced flawed protocols that allowed coercion: The Seller could force the Buyer into sending the payment without ever sending the physical good [Goh21]. Because of this design flaw, such protocols are not applicable in practice. There is a need for protocols that are **incoercible**, meaning that none of the involved parties has an advantage by not fulfilling its obligation.

1.1 Research Topic

Investigate the existence of a protocol that enables a trustless and incoercible sale of physical goods over a blockchain.

The commonly known protocols lacked at least one of the previously described important properties. Some protocols use a trusted third party, other protocols allow that one party coerces the other. There is no protocol that enables a incoercible sale of physical goods over a blockchain without a trusted escrow. This led us to make the investigation of such a protocol the topic of this master thesis.

We want to find a protocol where it is profitable for each party to not deviate from this protocol: Following the protocol is the only way for each party to maximize its utility. If such a protocol is found, as it is in the best interest of the parties to follow the protocol, it solves the problem of an incoercible sale without trust. The Seller can expect the Buyer to pay not because of trust, but because it is the most profitable option for the Buyer himself.

Looking forward, we do not find a two-party protocol that is a solution to the problem. As we will prove under reasonable assumptions, there cannot exist a Smart Contract that can implements a trustless and incoercible sale of a physical good over a blockchain. Still, we propose various solutions that introduce a third party that is not trusted by both parties and does not act like an escrow. We discuss to what degree there is a need of trust and the practicality of these solutions.

1.2 Methodology

First we formalize the existing problem of the research topic using a game-theoretic approach [LS08]. In order to find a solution to the problem, we propose various candidates. To find a solution that is as simple as possible but as complex as necessary, we take an iterative approach described in the following paragraph.

We initially propose the simplest possible solution and game-theoretically analyse if it indeed solves our problem. If the game-theoretic analysis shows that the proposed protocol is not a valid solution, we propose a new solution that has an added layer of complexity. This cycle is continued until we find a solution or we come to the conclusion that we have exhausted all reasonable solutions.

As this iterative approach does not lead us to a viable solution we later try to prove that there is no solution to the formalized problem. After the proof we change the requirements to the solution and thereby modify the problem. We allow third party, called the Mediator, to participate in the protocol. Afterwards we search for solutions to the newly defined problem. The following list describes our methodology approach:

1. Formalization of the problem
2. Iterative approach to find a solution to the problem
   a) **A solution is found:** This solution is the result of this thesis.
   b) **No solution is found:** Try to proof that no solution to the initial problem exists.
3. Loosening assumptions and define a modified problem
4. Return to step 2 with the new problem

1.3 Structure of this thesis

In chapter 2 we introduce the general concepts of game theory, cryptographic commitment schemes, blockchains and Smart Contracts. After that in chapter 3 we formalize the problem by defining the corresponding game.

In chapter 4 we propose numerous possible two-party protocols and analyse if these protocols are a viable solution to the proposed problem. Section 4.1 looks at protocols where no player deposits collateral, sections 4.2, 4.3, 4.4 examine protocols where only one party deposits collateral and section 4.5 looks at protocols where both parties deposit collateral. In Section 4.6 we show that for a protocol to solve the proposed problem, it needs an extra input which it is not given in the real world. This corresponds to step 2b of the methodology approach.

In chapter 5 we define the new game where we allow a third party to be a part of the protocol. For this new game we try to find solutions, that are applicable to real world scenarios. This corresponds to step 2a from our methodology approach. Finally, chapter 6 concludes this thesis and summarizes the results of this work. It also provides topic for further research.

1.4 Related work

Solving the Buyer and Seller's Dilemma: A Dual-Deposit Escrow Smart Contract for Provably Cheat-Proof Delivery and Payment for a Digital Good without a Trusted Mediator [AK19]
In this paper Asgaonkar and Krishnamachari propose a protocol that allows the sale of a digital good without a trusted third party. Their solution involves a Smart Contract that collects a deposit from both the selling and the buying party. If then one party misbehaves, the Smart Contract slashes the deposit of the lying party. This solution assumes the good to be digitally verifiable. Therefore if one party is not honest, the opposing honest party can report this misbehaviour to the Smart Contract and prove that the other party indeed did misbehave. The authors also provide a game-theoretic analysis of their protocol and confirm that honest behaviour is the only Nash equilibrium.

Two Party double deposit trustless escrow in cryptographic networks and Bitcoin. [Zim]
In this work Zimbeck proposes a protocol based on bitcoin scripts that should allow the trade between two perfect strangers even if the parties themselves cannot be trusted. His approach has no need for a trusted third party. Zimbeck also implemented his protocol in an open source program called BitHalo. His approach was formally and game-theoretically analysed in the paper [Big+15]. The protocol proposed by Zimbeck has the serious flaw that the Seller can coerce the Buyer[Goh21], making it not applicable in any sale of goods in the real world.

Escrow Protocols for Cryptocurrencies: How to Buy Physical Goods Using Bitcoin [Gol+17]
In this paper Goldfeder et al. look at the problem of buying physical goods using cryptocurrencies with a third party acting as an escrow. They formalized the escrow problem for physical goods and introduce multiple schemes with different properties.

Irrationality, Extortion, or Trusted Third-parties: Why it is Impossible to Buy and Sell Physical Goods Securely on the Blockchain [Goh21]
In this paper Goharshady analyse the sale of a physical good over a blockchain. They show that previous approaches like BitHalo (proposed in [Zim]) have the serious flaw that the Seller can coerce the Buyer into sending him the payment without ever receiving the physical good. In addition to that they provide a proof that any strictly two-party escrow contract for the sale of a physical good that is deployed between rational parties on a blockchain enables the Seller to extort the Buyer.

The work of Goharshady and our work look at a very similar problem. In particular the proof that they provided and the proof in section 4.6.3 prove similar things. The drawback of our approach is that the scope of our result is narrower, as we prove that there exists no Smart Contract that can the achieve the ideal functionality needed for the two-side collateral protocol to work. In contrast to that, Goharshady proves that any two-party protocol for the sale of a physical good is deployed between rational parties on a blockchain allows coercion. They have the additional assumptions that the Seller can always send messages to the Buyer (e.g. through a note in the sent package) and the Buyer will read this message and act upon it. Protocols like the mediated protocol introduced in section 5.3 break this assumption.

2 Preliminaries

2.1 Game Theory

Game theory studies the interaction between self-interested players [LS08]. It formalizes how a player behaves in a given game.

2.1.1 Games

A game consists of players who are taking actions in order to maximize their own payoff. An action is a choice that a player can make. Each possible action for a given decision is contained in an action set. Which choice the player makes is influencing how big his payoff will be. The payoff (or utility) is a numerical value and each player gets his own payoff. It is the goal of each player to maximize his own payoff. He will take every action that leads him to achieve this goal.

Definition 1 (Game)**.** *A game is a tuple (N, A, u), where:*

- *N is a finite set of n players.*
- *$A = A_1 \times \cdots \times A_n$, where A_i is a finite set of actions available to player i.*
- *$u_1, ..., u_n$ where u_i: $A \to \mathbb{R}$ is a payout (or utility) function for player i.*

[LS08]

2.1.2 Strategies

Each player faces multiple decisions in a game. At every decision point he chooses between possible actions. These decision can depend on every information that the player knows at this stage of the game. Which action a player takes in which situation can be formulated in a strategy. A possible strategy could be: If the opposing player takes action L, I will take action R, else I will take action L.

Definition 2 (Strategy)**.** *Player i's strategy s_i is a rule that tells him which action to choose at each instant of the game, given his information set.* [Ras].

We call the set of every strategy for a player his strategy set. A strategy profile is an ordered set consisting of one strategy for each of the n players in the game.

Definition 3 (Dominated strategy)**.** *The strategy s_i^d is dominated by s_i' if it is strictly inferior to s_i' no matter what strategies s_{-i}^* the other players choose, in the sense that whatever strategies they pick, his payoff is lower with s_i^d than with s_i'*[Ras]*. Formally:*

$$\forall s_{-i}^* : u_i(s_i^d, s_{-i}^*) \leq u_i(s_i', s_{-i}^*) \tag{2.1}$$

If one strategy dominated the other, a player will always choose the dominant strategy.

2.1.3 Nash Equilibrium

Definition 4 (Nash Equilibrium)**.** *A strategy profile s^* is a Nash equilibrium if no player has incentive to deviate from his strategy given that the other players do not deviate.* [Ras] *Formally:*

$$\forall i : \forall s_i' : u_i(s_i^*, s_{-i}^*) \geq u_i(s_i', s_{-i}^*) \tag{2.2}$$

Nash showed, that every game with a finite number of players, each having a finite set of strategies, has at least one Nash Equilibrium [Nas50].

2.1.4 Two-party alternating turn-based games

Definition 5 (Two-party alternating turn-based games)**.** *A game is a two-party alternating turn-based game if two parties are playing alternately, only one party can play at a time and its move is revealed to the other party before the other party plays the next move.*

For example, chess is a two-party alternating turn based game.

To analyse two-party alternating turn-based games we will use game trees. The game starts from the root of the tree and ends at the leaves of the tree. The nodes of the tree represent decision points. Each node is marked with the party that decides. The edges of the tree represent possible actions. At each node the party that plays chooses 1 action among the available edges that lead to the children of the node. The leaves of the tree are the payout values for the players. In this example Alice and Bob have one decision each. Alice will play first. She has the possible actions L and R. Bob will then make his decision choosing between the actions L and R.

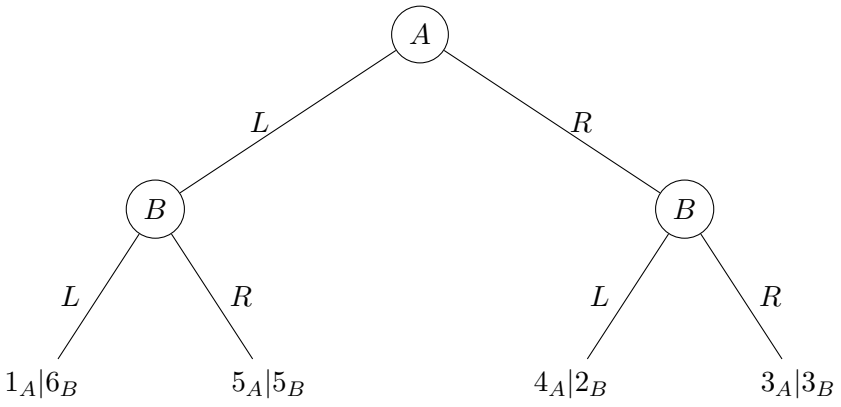


Figure 2.1: Example Tree

To find the Nash Equilibrium of this example game of Fig. 2.1 we will iteratively eliminate strictly dominated strategies [Ras]. We start at the bottom of the tree at Bob's decision. He always takes the action that yields him the biggest payout. We start at the left branch. If Bob takes action L, he gets a payout of 6. If he chooses action R, he gets a payout of 5. In this scenario Bob chooses L, because $6 > 5$. Likewise, on the right branch he chooses R because $3 > 2$.

Then we look at Alice's decision. She knows that Bob chooses the path that yields to him the greatest payout and can calculate his decisions. She knows that Bob will choose L,if Alice takes action L. Therefore her payout of action L is 1. If she chooses action R, Bob will choose R as well. In this case Alice has a payoff of 3. Alice chooses action L, because $3 > 1$.

The Nash Equilibrium of this game is $\{R,R\}$ and the payout is $3_A|3_B$. It is interesting to see, that this is for both parties a worse outcome than $\{L,R\}$, where the payoff is $5_A|5_B$.

2.1.5 Two-party simultaneous games

Definition 6 (Two-party simultaneous games)**.** *A game is a two-party simultaneous game, if the two parties choose their strategy simultaneously and independently from each other and both parties have only one decision to make. After that, all actions are revealed and the relevant payoffs are given.*

For example, rock-paper-scissors is a two-party simultaneous game.

To analyse two-party simultaneous games we will use a table. The two players are again Alice and Bob. On the top of the table we see the possible strategies of Bob: *be honest* and *lie*. On the left side are the possible strategies for Alice: *report* and *hide*. For each combination of strategies the two parties get a specific payout. For example if Bob chooses strategy *lie* and Alice chooses strategy hide, Bob gets a payout of 8 and Alice gets a payout of 2.

Alice \ Bob	*be honest*	*lie*
report	2 | 5	5 | −3
hide	1 | 3	2 | 8

Table 2.1: Example Table

If we now look at the table we can see which combinations of strategy benefits which player. There is a Nash Equilibrium at {*be honest*, *report*}, because no player can improve their payout by changing strategy as long as the other party does not change its strategy. If Bob changes his strategy to *lie* and Alice still chooses *report*, Bob's payout will decrease to −3. If a Alice changes her strategy to *hide* and Bob still chooses *be honest*, her payout will decrease to 1.

In every two-party simultaneous game with a Nash Equilibrium at {*NE_column*, *NE_row*}, the party that chooses the column (Bob in the example) will always choose the maximum of its own payouts of the row *NE_row*. In the example, Bob chooses the column with the biggest blue value of the row *report*, which is *be honest*. Likewise, the player that chooses the row will choose the row that maximizes its own payoff of column *NE_column*. In the example, Alice chooses the row with the biggest red value in the column *be honest*, which is *report*.

2.2 Commitment schemes

Commitment schemes are used in a vast variety of cryptographic protocols. They enable a party to commit itself to a value while keeping it secret [Gol01]. This resembles a non transparent sealed envelope: By putting a note in such an envelope, a party commits itself to the content of the note. This note can not be read until the envelope is being opened.

A commitment scheme lets one party (Bob) commit to a value of his choice. Later, he can open the commitment in order to convince another party (Alice) that indeed that value was committed. Committing to a value results in a commitment string c, that Bob sends to Alice, and an opening string s that Bob uses for opening the commitment later[BS20].

A commitment scheme is secure if it satisfies the following two properties:

- **Hiding:** The Commitment string c reveals no information about the committed value[BS20].
- **Binding:** Let c be a commitment string output that Bob gets by committing the value v. Bob can open this commitment to the committed value v, but not to any other value $v' \neq v$.

2.3 Blockchains

To understand blockchains and their origin, we first look at cryptocurrencies. Cryptocurrencies are decentralised currencies that introduce a way to transfer monetary value with low latency and without any intermediary. They use various cryptographic primitives to prevent fraud. For example, they use digital signature to ensure that only the owner of the coin can spend it and nobody else. In addition to that another important cryptographic primitive are Hash Functions. A Hash Function can take an arbitrary large file as input and output a short string, a *hash* of constant size. A detailed description of said cryptographic primitives can be found at Katz and Lindell's book *Introduction into modern Cryptography* [KL20].

We will explain some of the principles of cryptocurrencies by looking at the first and best known cryptocurrency: Bitcoin. Bitcoin was introduced in 2008 [Nak08]. In Bitcoin everyone can create his own wallet which he can use to send and receive coins. These coins are sent using *transactions*. Every transaction is stored on a public distributed database, the *blockchain* (c.f. section 2.3). Once these transactions are on the blockchain, they can never be changed or deleted. The blockchain acts as a public ledger recording all the transactions. These transactions are grouped in blocks. Aside from transactions, each block contains the hash of the previous block, thereby linking it with the previous block, forming a chain of blocks.

New transactions are written in new blocks that are appended to the blockchain. These blocks are created by the *validators*. The validators are very important for the security of the blockchain, as they define the unique ordered history of transactions. They are usually a big, dynamic set of people, meaning that the set of active validators may change over time. In addition to that, some percentage of this set of validators may be arbitrarily malicious. Even in this case, the blockchain has to guarantee its security properties.

The security properties a blockchain has to fulfil in order to be a public transaction ledger are [GKL15]:

- **Common Prefix:** Honest validators have identical prefix with high probability, meaning that honest validators agree on the same transaction history.
- **Chain Quality:** The blockchain of honest validators contains a significant fraction of blocks created by honest validators.
- **Chain Growth:** After a certain time interval it is guaranteed that new blocks are added to the blockchain.

After Bitcoin many other cryptocurrencies were developed. In this work, we do not look at a specific cryptocurrency. We rather use them as a remote way to transfer monetary value between two parties. These two parties do not need to know each other, rather they only need each other's wallet address to send coins to each other. The transfer of value occurs nearly instantaneous and can not be reversed.

2.4 Smart Contracts

Smart contracts are small computer programs that are stored and executed on the blockchain [CD16]. They have their own on-chain address, the ability to own coins and they can interact with parties and other Smart Contracts via transactions. Anybody can create a Smart Contract with a transaction by uploading its program code to the blockchain [Nar+16]. As Smart Contracts are self-executing and tamper resistant [MPJ18], they can be used to enforce an agreement without a third party.

The concept of Smart Contract can be illustrated by the following example: Alice and Bob want to make following agreement: Alice thinks of a word. She will pay Bob 5 coins, if he guesses the word correctly. If not, Bob will pay Alice 2 coins. As they do not trust each other, this agreement is difficult

to enforce traditionally without an independent party. A Smart Contract could help Alice and Bob enforce this agreement: Alice creates the Smart Contract and sends 5 coins to it with a transaction. Then Bob sends 2 coins to the Smart Contract. After that Alice sends another transaction with her secret word (Note that all transactions are public, so it is better to send the hash of the word in order to prevent Bob from seeing this word). Bob can then send his guess in another transaction. If he guesses correctly, the Smart Contract automatically rewards Bob all 7 coins, else it rewards Alice all 7 coins.

Smart Contracts were popularized by Ethereum [But14], the second biggest cryptocurrency after Bitcoin. In Contrast to Bitcoins scripts, Smart Contracts in Ethereum are Turing-complete and therefore have a lot more functionality.

3 Formalization

Definition 7 (Utility of Physical Good)**.** *The utility of a physical good $U_i(G)$ describes how much utility a physical good G provides to a player i. The output of this function is a numerical value, which corresponds to the amount of coins that have equal value as $U_i(G)$. In this work, utility will always be a positive integer.*

3.1 Entities

Definition 8 (Seller)**.** *A rational entity S that initially owns a specific physical good G. The Seller has bigger utility in owning P coins in contrast to owning G (c.f. equation 3.1) He therefore desires to exchange G against P coins in order to maximize his utility.*

$$U_S(G) + P > P > U_S(G) > 0 \tag{3.1}$$

Definition 9 (Buyer)**.** *A rational entity B that initially owns at least P coins. The Buyer has greater utility in owning G in contrast to owning P coins (c.f. equation 3.2). He therefore desires to exchange P coins against G in order to maximize his utility.*

$$U_B(G) + P > U_B(G) > P > 0 \tag{3.2}$$

Definition 10 (Smart Contract)**.** *An Entity SC that is a Smart Contract on a blockchain. It is therefore trusted by the other entities. The SC interacts with the Buyer and the Seller by it taking the money and possibly collateral. It has the purpose to facilitate the exchange.*

Since the Smart Contract lives on the blockchain, it cannot hold on to the physical good. It has no information whether the physical good was sent by the Seller and received by the Buyer.

3.2 The Game

Definition 11 (Initial State)**.** *A Seller and a Buyer have some way of communicating with each other. Both parties have enough funds in the given cryptocurrency to carry out protocol actions: Both have enough collateral (if needed) and the Buyer has at least P additional coins. Both the Buyer and the Seller have access to the same Smart Contract. The Seller is in possession of the physical good.*

Definition 12 (Desired Final State)**.** *The Buyer is in possession of the physical good and his funds have decreased by amount P. The funds of the Seller have increased by P.*

Definition 13 (Success of the trade)**.** *The trade is considered successful if for all parties starting in the Initial State, the Desired Final State is a Nash Equilibrium.*

The goal of this work is to design a protocol that enables a successful trade as defined in definition 13. Minimal trust should be required. This will be achieved, if it is the rational behaviour for both players to perform the trade. It should be a Nash Equilibrium to follow the protocol: If the other party follows the protocol, the best strategy will be to follow the protocol as well.

Another goal is that the trade is incoercible against rational attacks. That means that it should be unprofitable to attack and decrease the payout of the other player, if the other player is following the

protocol. If this goal is achieved, a rational player will never try and attack the opposing player. It should be noted that an irrational player can attack the opposing party, as an attack is unprofitable, but not necessarily impossible. If a player is willing to lower his own utility, he is able to attack an opposing player.

We make the following assumptions:

1. All entities have access to the Smart Contract and know the addresses and public keys of each other.
2. The Seller knows the physical address to which the Buyer wants the good to be shipped. This address is not necessarily the Buyer's real physical address, he can choose any physical address he wants to protect his privacy.
3. Other than the information of assumption 1 and 2, the Buyer and the Seller know nothing about each other.
4. Both parties can send funds in the given cryptocurrency to any address.
5. The sender has the power to send the physical good to any physical address he wants.
6. Both the Seller and the Buyer know and trust the code of the Smart Contract.
7. The Seller and the Buyer behave rationally, meaning that they both want to maximise their expected payout.
8. Both $U_S(G)$ and $U_B(G)$ are positive.

We do not take into account transaction fees.

4 Candidate Two-Party Protocols

In this chapter we try to find a two-party protocol that enables a trustless and incoercible sale over a blockchain. We start by a simple, naive approach and then increase the complexity on each proposed protocol. On each protocol we examine the created game and analyze the Nash Equilibrium of that game. As described in section 3.2, we want to create a game where the Nash Equilibrium lies in an successful trade between the Seller and the Buyer.

4.1 First try: no collateral

The simplest protocol is both players sending their resources without any collateral nor a third party. In this case there are 3 possibilities:

Buyer sends first	
1 :	Buyer sends P coins to the Seller
2 :	Seller sends the physical good G to the Buyer

Figure 4.1: Protocol: No collateral, Buyer sends first

Seller sends first	
1 :	Seller sends the physical good G to the Buyer
2 :	Buyer sends P coins to the Seller

Figure 4.2: Protocol: No collateral, Seller sends first

Both players send simultaneously	
1 :	Buyer sends P coins to the Seller and Seller sends G to the Buyer simultaneously

Figure 4.3: Protocol: No collateral, both players send at simultaneously

4.1.1 Possible strategies

There is a countless amount of possible strategies for the two players. There are many examples for possible strategies: A player can follow the protocol and trade successfully, he can try and gain an advantage by not following the protocol, he can ignore all messages or he can send all his money to the other party, and many more. It is not sensible to list and analyse all the possible strategies. We will concentrate on the strategies where a player follows the protocol and the strategies where a player intentionally deviates from the protocol to gain an advantage. As soon as we find a strategy for a player that is more profitable than following the protocol, we know that this protocol is not achieving our goals, as a rational player will not follow it. If we conclude that following the protocol is the only way for each player to maximize his utility, we know that this protocol is achieving our goal.

For the protocols proposed in section 4.1 we analyse two strategies for each player. The Buyer can choose between strategy *Pay* where he sends the money and strategy *Withhold* where the Buyer does

not send the money. The Seller can choose as well between strategy *Send* where he sends the good G and strategy *Keep* where he does not send the G.

4.1.2 Game: The Buyer acts first

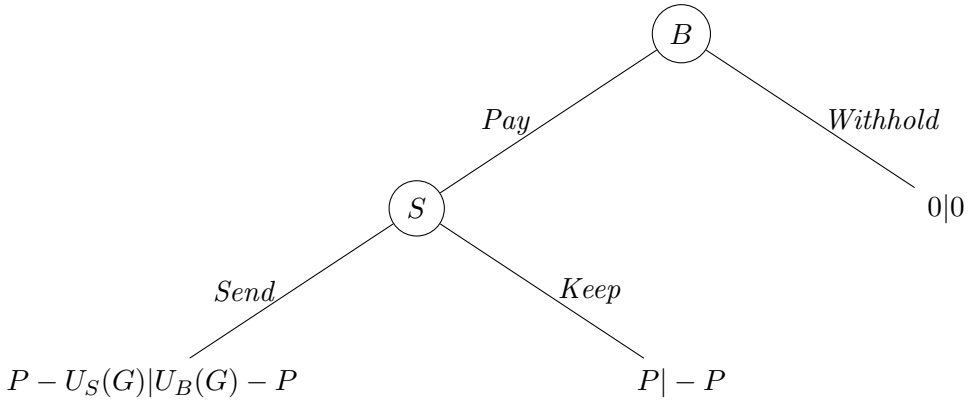


Figure 4.4: No collateral, Buyer acts first

The left value of each leaf is the payoff to the Seller and the right value of the leaf is the payoff to the Buyer. To analyze the game, we iteratively eliminate strictly dominated strategies (c.f. section 2.1.4). We start at the bottom at the Seller's decision.

- The Seller chooses *Keep* (Because $P > P - U_S(G)$).

We then go one step towards the root and look at the Buyer's decision. He already knows that if he chooses *Pay*, the rational Seller will choose *Keep*. This would leave the Buyer with a payout of $-P$. If the Buyer chooses *Withhold*, his payout is 0.

- The Buyer chooses *Withhold* (Because $0 > -P$)

The Nash Equilibrium of this game is $\{Keep \mid Withhold\}$ which is the initial state and results in a payout of 0|0. Both parties are not sending anything and the trade is unsuccessful.

4.1.3 Game: The Seller acts first

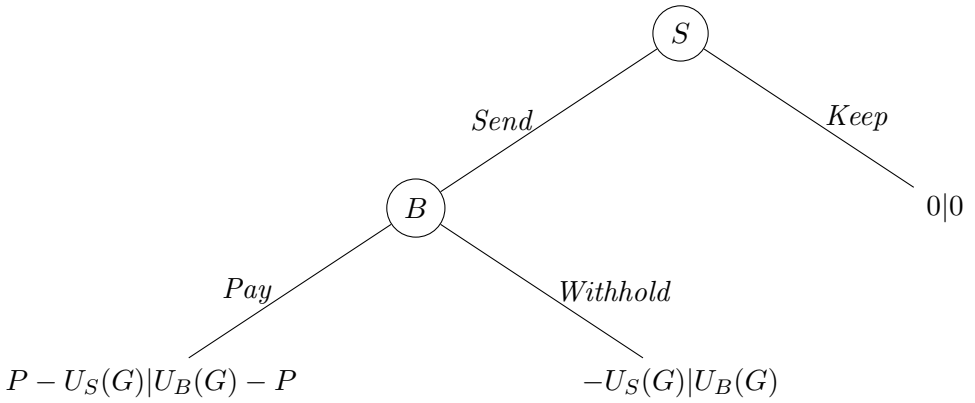


Figure 4.5: No collateral, Seller acts first

The analysis is similar to the above case, but the roles are reversed.

- The Buyer chooses *Withhold* as $U_B(G) > U_B(G) - P$

- The Seller chooses *Keep* as $0 > -U_S(G)$

The Nash Equilibrium of this game is $\{Keep \mid Withhold\}$ and results in a payout of $0|0$. Again, both parties are not sending anything and the trade is unsuccessful.

4.1.4 Game: Both parties act simultaneously

Seller \ Buyer	*Pay*	*Withhold*
Send	$P - U_S(G) \mid U_B(G) - P$	$-U_S(G) \mid U_B(G)$
Keep	$P \mid -P$	$0 \mid 0$

Table 4.1: No collateral: Both parties send simultaneously

If we look at the payout table we can see that *Withhold* dominates *Pay* and *Keep* dominates *Send*. These strategies provide a bigger payout regardless of the strategy of the other party. Since we assume the players to be rational, they would both choose the dominant strategy. Again, the Nash Equilibrium of this game is $\{Keep \mid Withhold\}$ and results in a payout of $0|0$. As the Nash Equilibrium does not correspond to the Desired Final State (c.f. Definition 12), this game does not satisfy our goal.

It is interesting to note that this table resembles the famous prisoner's dilemma. We can see that the Nash Equilibrium $\{Keep \mid Withhold\}$ is not the optimal case for both parties, because $\{Send \mid Pay\}$ increases the payout of both parties. In other words, a successful trade would benefit both parties but is not a Nash Equilibrium in this game.

4.2 Second Try: Buyer Collateral

To incentivize an entity to follow the protocol we introduce collateral. The collateral will be lost if the entity deviates from the protocol. In this section we will look at a protocol where only one the Buyer deposits collateral. We introduce a Smart Contract (SC) which handles the collateral. Instead of sending the Payment P directly to the Seller, the Buyer will send the payment to the Smart Contract. The Smart Contract will then forward the payment to the Seller. This hands more power to the Smart Contract. With control over the collateral and the payment it has more options to punish misbehaving parties.

The following protocol is proposed:

Buyer collateral protocol	
1 :	Buyer deposits collateral $C > P$ to the Smart Contract
2 :	Seller sends the physical good G to the Buyer
3 :	After receiving G, the Buyer sends funds of amount P to the Smart Contract
4 :	The Smart Contract releases the collateral of the Buyer and sends P to the Seller

Figure 4.6: Buyer collateral protocol

The Smart Contract releases the collateral as soon as it receives P by the Buyer. By paying P, the Buyer implicitly reports to the Smart Contract that he has received the physical good G. The collateral should incentivize the Buyer to pay the Seller after receiving the physical good G, as his collateral is greater than P.

It is important that the Seller sends first, as he has no collateral and therefore no incentive to follow the protocol. If we swap the order and let the Buyer send first, the Seller will simply keep G and be

in possession of both G and P. Because of the collateral, the Seller can safely send G first without having to worry that the Buyer will not pay him.

4.2.1 Possible Strategies

We still have the strategies from section 4.1.1: *Pay* and *Withhold* for the Buyer and *Keep* and *Send* for the Seller. The Buyer now has one additional decision: He can either follow the protocol and deposit the collateral C (*Deposit*) or deviate from the protocol and refuse to deposit the collateral (*Refuse*).

4.2.2 Game

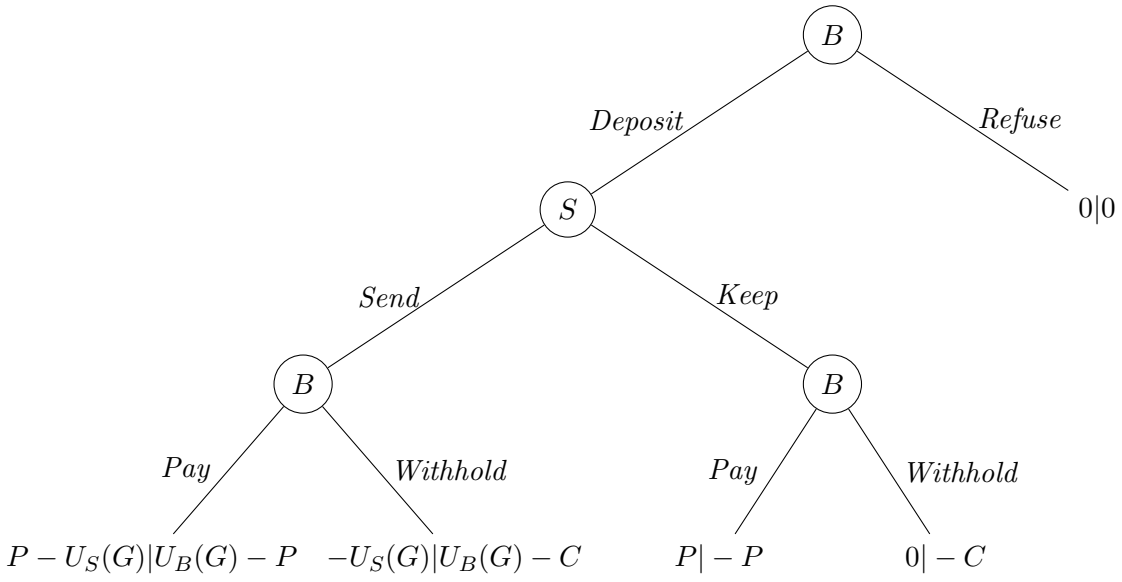


Figure 4.7: Only the Buyer deposits collateral

To analyze the payout of this game, we again start at the bottom at the Buyer's decision. On both branches the Buyer chooses *Pay*:

- $U_B(G) - P > U_B(G) - C$
- $-P > -C$

We now look at the Seller's decision. He chooses *Keep*:

- $P > P - U_S(G)$

This means that it is better for the Seller to deviate from the protocol and not send the physical good G. The Buyer would send P just to get back his collateral but receive nothing more, leaving him with a payout of $-P$. At the root of the tree the Buyer chooses $Refuse$:

- $-P < 0$

The Nash Equilibrium is at $\{Keep \mid Refuse,\ Pay\}$ and results in a payout of $0|0$. The trade is unsuccessful.

4.3 Third Try: Seller Collateral

In this section we will look into protocols where only the Seller deposits collateral.

The following protocol is proposed:

Seller collateral protocol	
1 :	Seller deposits collateral $C > P$ to the Smart Contract
2 :	Buyer sends funds of amount P to the Seller
3 :	Seller sends the physical good to the Buyer
4 :	Buyer receives the physical good and confirms it to the Smart Contract
5 :	The Smart Contract releases the collateral of the Seller

Figure 4.8: Seller collateral protocol

The collateral should incentivize the Seller to follow the protocol and ship the physical good to the Buyer. The Buyer can report to the *SC* with a simple boolean input:

- 0 denotes that G was not received
- 1 denotes that G was received

4.3.1 Possible Strategies

There are still the basic strategies from section 4.1.1: *Pay* and *Withhold* for the Buyer and *Keep* and *Send* for the Seller. Similar to the strategies of the Buyer Collateral protocol (c.f. 4.2.1) the Seller has the option to deposit the collateral (*Deposit*) or refuse to deposit the collateral (*Refuse*).

The Buyer has a unique opportunity: He decides what happens to the Seller's collateral as he can either report that he received the physical good G (*Report*) or he can report that he did not receive it. Because he has this power over the Seller's collateral, he can try and coerce the Seller into sending him extra resources. This strategy will be denoted as *Coerce*. After receiving the physical good he communicates to the Seller: "I will only report that I received the good if you send me back my payment P".

It is important to see that this is a credible threat, because the Buyer's payout is not affected by him reporting or not reporting. Still, he has to find a way to convince the Seller that this statement is true. For the coercion to work, the Seller has to be convinced that he will indeed receive back his collateral if he sends the Payment P back to the Seller. One possible way of convincing the Seller is via a second Smart Contract set up by the Buyer. As soon as the payment P is sent back to the Buyer, this second SC will automatically report 1 to the first Smart Contract and the Seller's collateral is released.

If the Buyer chooses *Coerce*, the Seller has two possible actions: He can comply and send back P (*Comply*) or not comply and ignore the message (*Ignore*).

4.3.2 Game

To improve readability, step 1 of the protocol is omitted from all trees in section 4.3 and 4.4. If the Seller does not deposit the collateral, the Smart Contract aborts the trade and the payout is $0|0$.

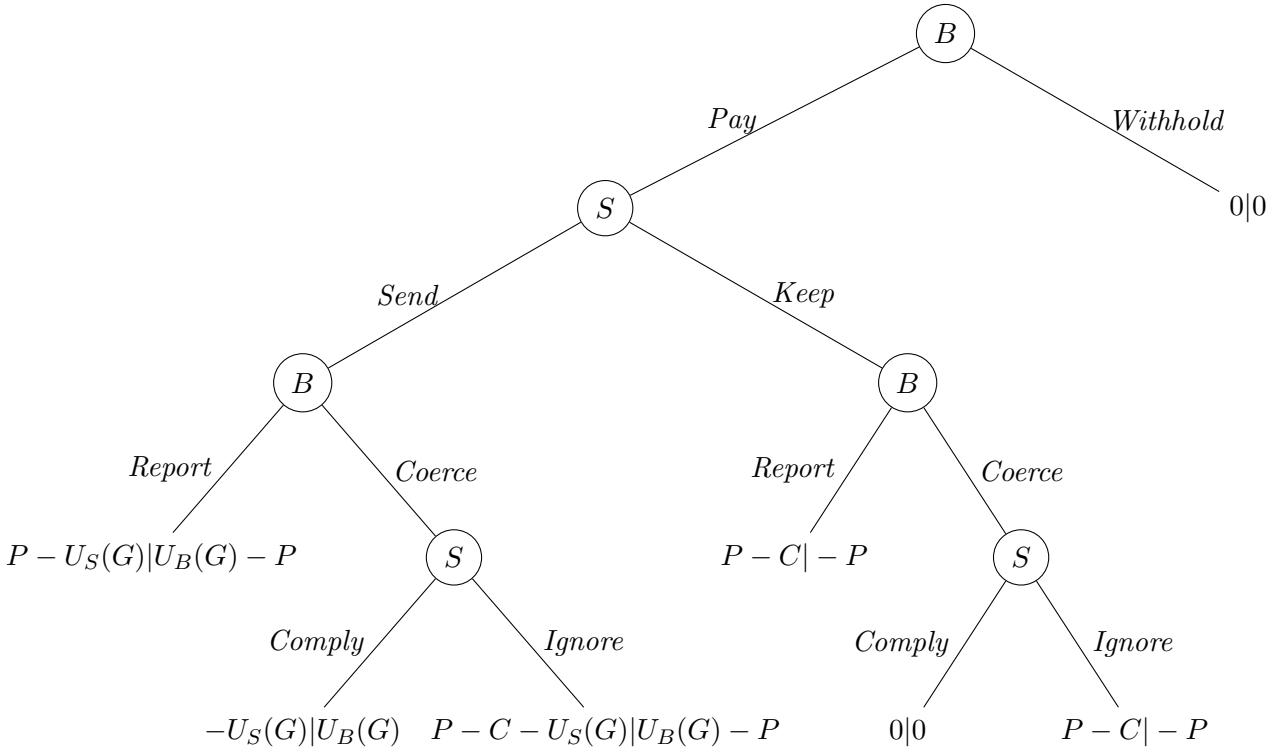


Figure 4.9: Only the Seller deposits collateral

To analyze the payout, we again start at the bottom. Since $C > P$, The Seller would choose *Comply* on both instances:

- $-U_S(G) > P - C - U_S(G)$
- $0 > P - C$

One step above, the Buyer chooses *Coerce* on both instances:

- $U_B(G) > U_B(G) - P$
- $0 > -P$

Based on this information the Seller chooses *Keep*:

- $0 > -U_S(G)$

That means that both options for the Buyer would result in a payout of 0|0. It does not matter which option he chooses. The Nash Equilibrium is at {*Keep*, *Comply* | *Withhold*, *Coerce*} and results again at a payout of 0|0. The trade is not successful. What was designed to be an incentive for the Seller to follow the protocol, ended up as leverage for the Buyer over the Seller. The Buyer can report 0 (and burn the collateral) even if S follows the protocol.

4.4 Fourth try: Seller collateral, both parties report

One Side Collateral on the Seller's side does not work if only the Buyer reports to the Smart Contract (c.f. section 4.3). If only the Buyer reports, he has all the power and can report anything he wants to the Smart Contract. Another option would be that both players report to the Smart Contract whether the physical good G was sent and received. If the two players report a different value, then the Smart Contract will slash the collateral of both parties as one party has to be lying. Their collateral is only returned if both parties agree and report the same answer to the Smart Contract. If they both report 0 (G was not sent), then the Smart Contract will send P back to the Buyer, because G was not shipped. If they both report 1 (G was sent), then the Smart Contract will send P to the Seller and the trade will be successful.

The following protocol is proposed:

Seller collateral protocol: Both parties report	
1 :	Seller deposits collateral $C > P$ to the Smart Contract
2 :	Buyer sends funds of amount P to the Smart Contract
3 :	Seller sends the physical good to the Buyer and the Buyer receives it
4 :	Buyer and Seller report the physical transaction to the Smart Contract
5 :	The Smart Contract releases the collateral of the Seller and sends P to the Seller

Figure 4.10: Seller collateral protocol: Both parties report

There are multiple ways to implement step 4: Either the two parties report one after another or they report simultaneously.

4.4.1 Possible Strategies

There are still the same strategies as in section 4.3.1. We now denote reporting 0 to the SC as "0" and reporting 1 to the SC as "1".

4.4.2 Game: The Seller reports first

As the Buyer reports last, this protocol still gives the Buyer the power of burning the collateral of the Seller by reporting exactly the opposite from the Seller's report. Therefore he can still coerce the Seller into sending him back his payment. As we already have shown in section 4.3, a rational Seller will always comply and send back the payment P. This last decision from the Seller (leaf 3, 6, 9, 12) will be omitted from the tree for better readability.

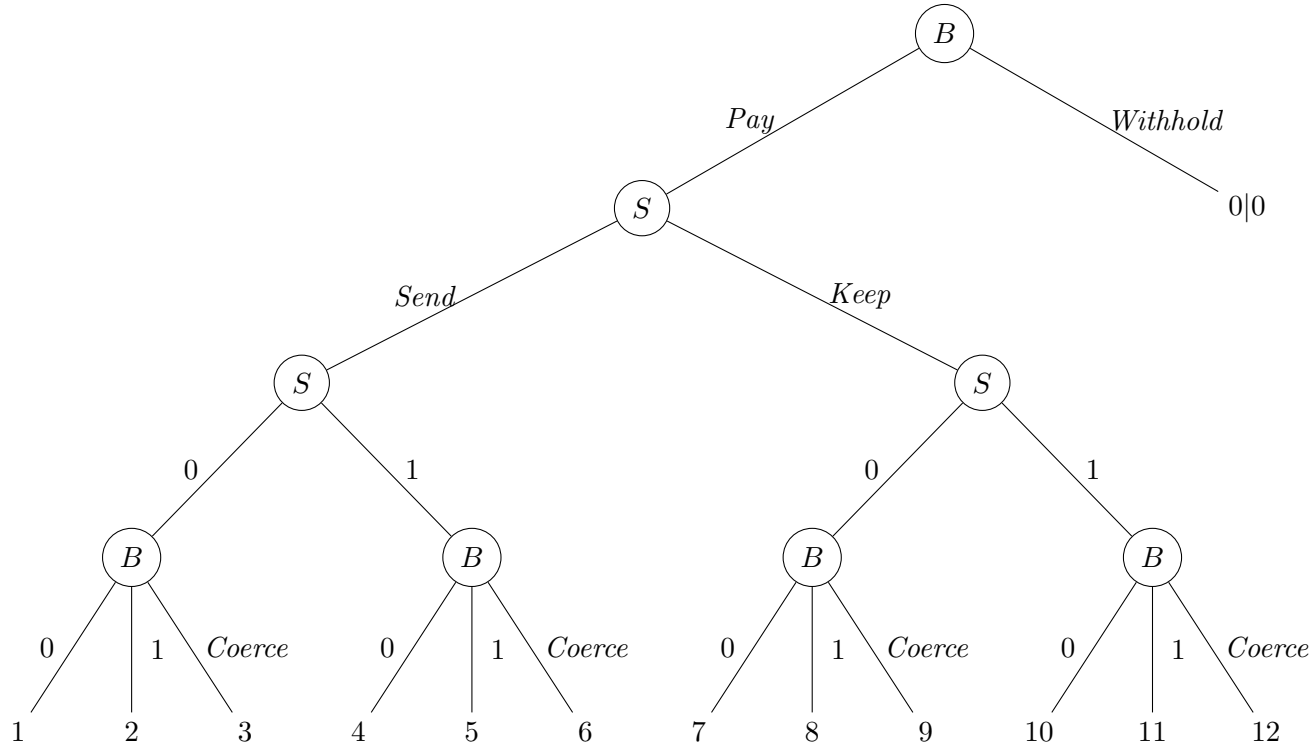


Figure 4.11: Seller collateral protocol: Both parties report

	Payout
1	$-U_S(G) \mid U_B(G)$
2	$-C - U_S(G) \mid U_B(G) - P$
3	$-U_S(G) \mid U_B(G)$
4	$-C - U_S(G) \mid U_B(G) - P$
5	$P - U_S(G) \mid U_B(G) - P$
6	$-U_S(G) \mid U_B(G)$
7	$0 \mid 0$
8	$-C \mid -P$
9	$0 \mid 0$
10	$-C \mid -P$
11	$P \mid -P$
12	$0 \mid 0$

Table 4.2: The corresponding payout table of figure 4.11

To analyze the payout, we again start at the bottom. The Buyer chooses:

- 1 or 3 over 2 ($U_B(G) > U_B(G) - P$)
- 6 over 5 and 4 ($U_B(G) > U_B(G) - P$)
- 7 or 9 over 8 ($0 > -C$)
- 12 over 10 and 11 ($0 > -P$)

The Buyer chooses:

- 7 or 12 or 9 (all 0) over 1 and 6 and 3 ($0 > -U_S(G)$)

No matter which decision the Buyer takes at the top of the tree, the payout will be $0|0$ and the trade unsuccessful.

4.4.3 Game: The Buyer reports first

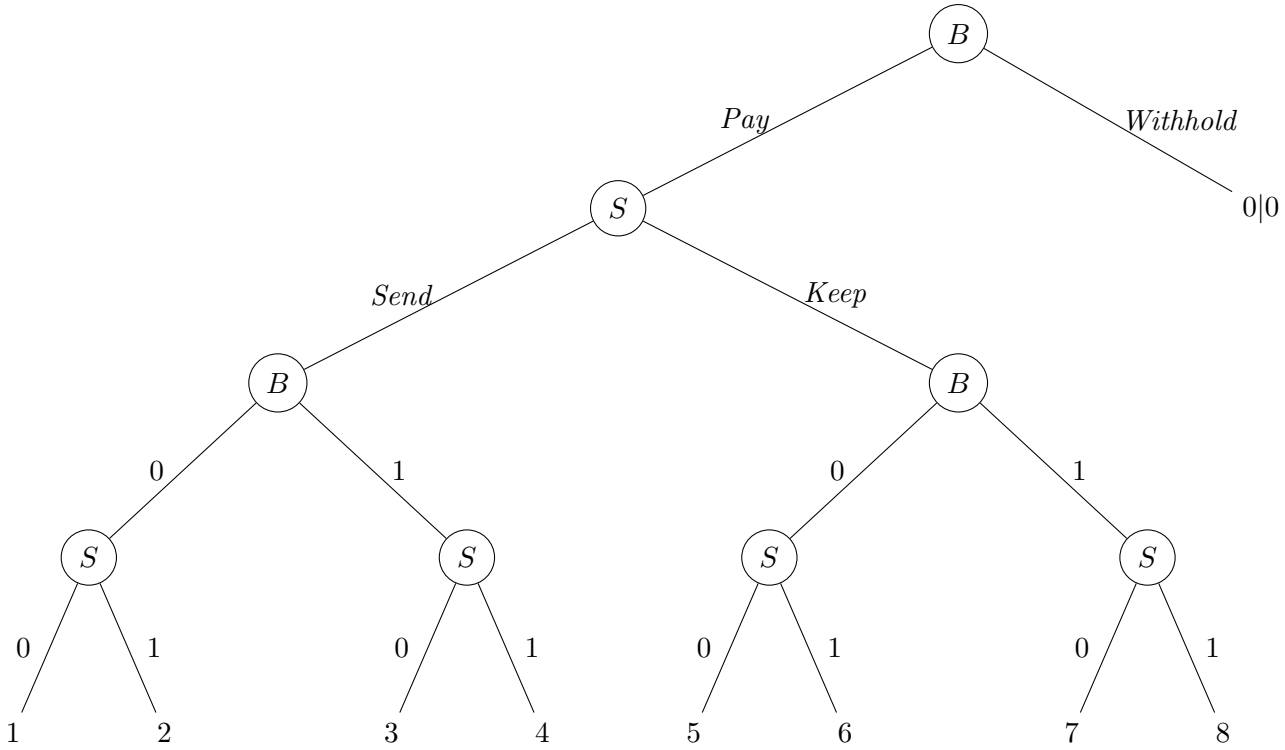


Figure 4.12: Only the Seller deposits collateral: Buyer reports first

	Payout
1	$-U_S(G) \mid U_B(G)$
2	$-C - U_S(G) \mid U_B(G) - P$
3	$-C - U_S(G) \mid U_B(G) - P$
4	$P - U_S(G) \mid U_B(G) - P$
5	$0 \mid 0$
6	$-C \mid -P$
7	$-C \mid -P$
8	$P \mid -P$

Table 4.3: The corresponding payout table of figure 4.12

To analyze the payout, we again start at the bottom. The Seller always reports the same as the Buyer, because he will lose his collateral if he reports anything different:

- 1 over 2 ($-U_S(G) > -C - U_S(G)$)
- 4 over 3 ($P - U_S(G) > -C - U_S(G)$)
- 5 over 6 ($0 > -C$)
- 8 over 7 ($P > -C$)

On the next decision the Buyer always reports that he did not receive the item:

- 1 over 4 ($U_B(G) > U_B(G) - P$)
- 5 over 8 ($0 > -P$)

Based on this information the Seller chooses *Keep*:

- $0 > -U_S(G)$

No matter which decision the Buyer takes at the top of the tree, the payout will be $0|0$ and the trade unsuccessful. **[Nikos: Nash: {*Keep, 1 if Buyer reported 1, else 0* | *Withhold, 0*}]**

4.4.4 Game: Both parties report simultaneously

How to report simultaneously

Every transaction to the Smart Contract is public. As the players have to report to the Smart Contract by sending a public transaction, the other player also knows what the first player reported. By just demanding both to report at the same time, the player who reports second always knows what value the first player reported. Then the report is not really simultaneous.

We can fix this problem by using a computationally hiding and binding commitment scheme. Both players would commit to their decision and send the commitment to the SC. The second player sees the commitment of the first player. Because the commitment scheme is computationally hiding and we assume the players to be computationally bounded, the commitment does not reveal any information about the actual decision of the first player. After both players send the commitment to the SC, they both send the opening value for their commitment to the Smart Contract. As the commitment is computationally binding, the players cannot change what they initially reported. If a player does not send an opening value after a predefined time, the SC punishes the player by sending all the funds to the other player.

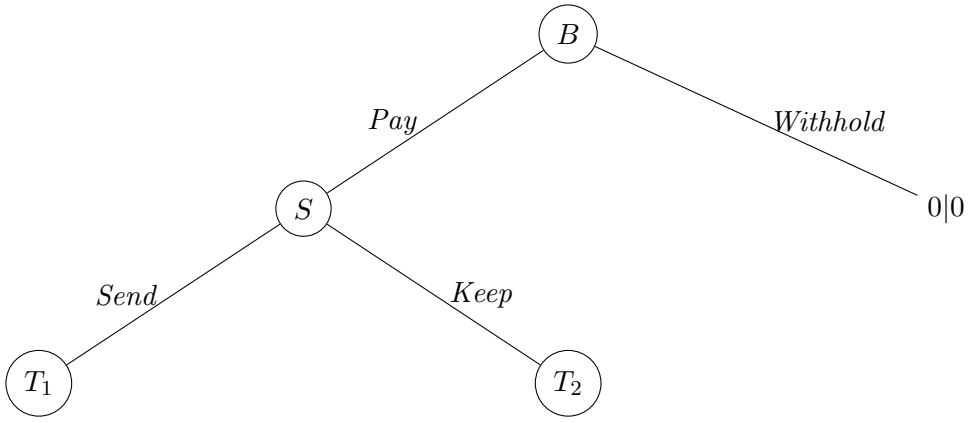


Figure 4.13: Only the Seller deposits collateral: Both players report simultaneously

Subgame T_1

Seller \ Buyer	0_B	1_B
0_S	$-U_S(G)\|U_B(G)$	$-U_S(G)-C\|U_B(G)-P$
1_S	$-U_S(G)-C\|U_B(G)-P$	$P-U_S(G)\|U_B(G)-P$

Table 4.4: Subgame T1: Both players report simultaneously

The Nash Equilibrium of subgame T_1 is $\{0_B, 0_S\}$ and payout $-U_S(G)|U_B(G)$.

Subgame T_2

Seller \ Buyer	0_B	1_B
0_S	$0\|0$	$-C\|-P$
1_S	$-C\|-P$	$P\|-P$

Table 4.5: Subgame T2: Both players report simultaneously

The Nash Equilibrium of subgame T_2 is $\{0_B, 0_S\}$ and payout $0|0$.

After considering the Nash Equilibria of both subgames the Seller chooses *Keep*:

- $0 > -U_S(G)$

As both options lead to a payout of 0, B chooses an arbitrary option. The trade is unsuccessful again.

4.5 Fifth Try: Two side collateral

As we saw in the previous three sections, one side collateral does not lead to a game that achieves our goal. The party that does not deposit collateral always has an unfair advantage over the other party. When only the Buyer deposits collateral, the Seller has no incentive to send G. When only the Seller deposits collateral, the Buyer can coerce the Seller into sending back his payment P, because the Buyer can burn the Seller's collateral by reporting 0.

We introduce another protocol that addresses these problems. In this protocol both parties deposit collateral. This should incentivize both parties follow the protocol. If the two parties report something different, the Smart Contract will slash the collateral of both players, as it can not determine which party is lying. The parties will get back their collateral only if they report the same value. If both parties report 0, the Smart Contract will send the payment P back to the Buyer. If both parties report 1, the Smart Contract will send the payment P to the Seller.

Again, there are three possible ways of handling the reports:

- The Seller reports first
- The Buyer reports first
- Both players report simultaneously

4.5.1 Possible strategies

There are still the basic strategies from section 4.1.1: *Pay* and *Withhold* for the Buyer and *Keep* and *Send* for the Seller. In addition to that the players can either report 0 or 1 to the SC, similar to section 4.4.

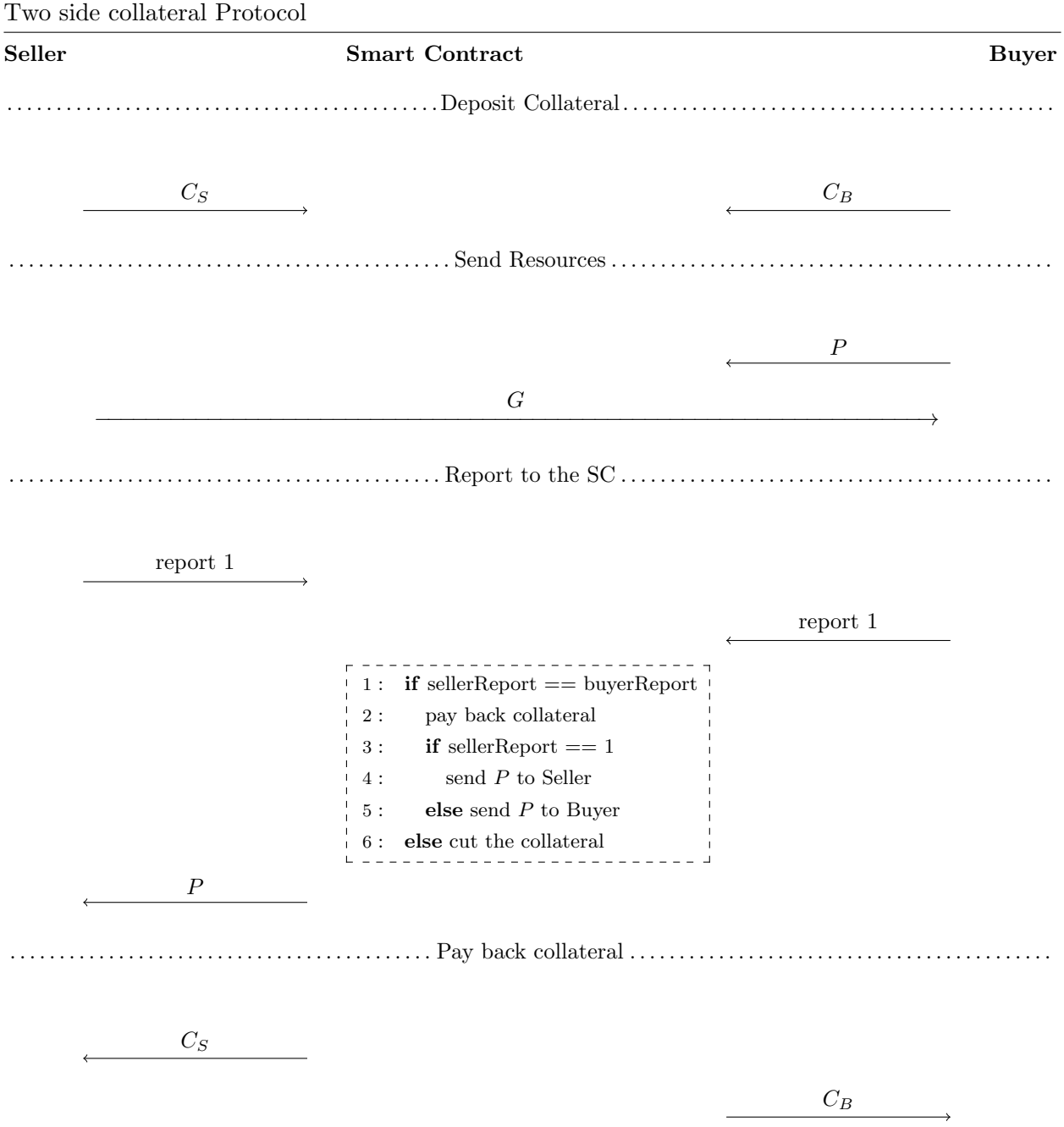


Figure 4.14: Two side collateral protocol

4.5.2 Game: The Seller reports first

The following protocol is proposed:

Two side collateral: Seller reports first	
1 :	Seller deposits collateral $C_S > P$ to the Smart Contract
2 :	Buyer deposits collateral $C_B > P$ and P to the Smart Contract
3 :	Seller sends physical good G to the Buyer and reports it to the Smart Contract
4 :	Buyer receives the physical good and reports it to the Smart Contract
5 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 4.15: Two side collateral protocol: Seller reports first

To increase readability, we do not plot the decisions from point 1 and point 2 of the protocol. If someone does not deposit the collateral, the Smart Contract aborts the trade and the payout is 0|0.

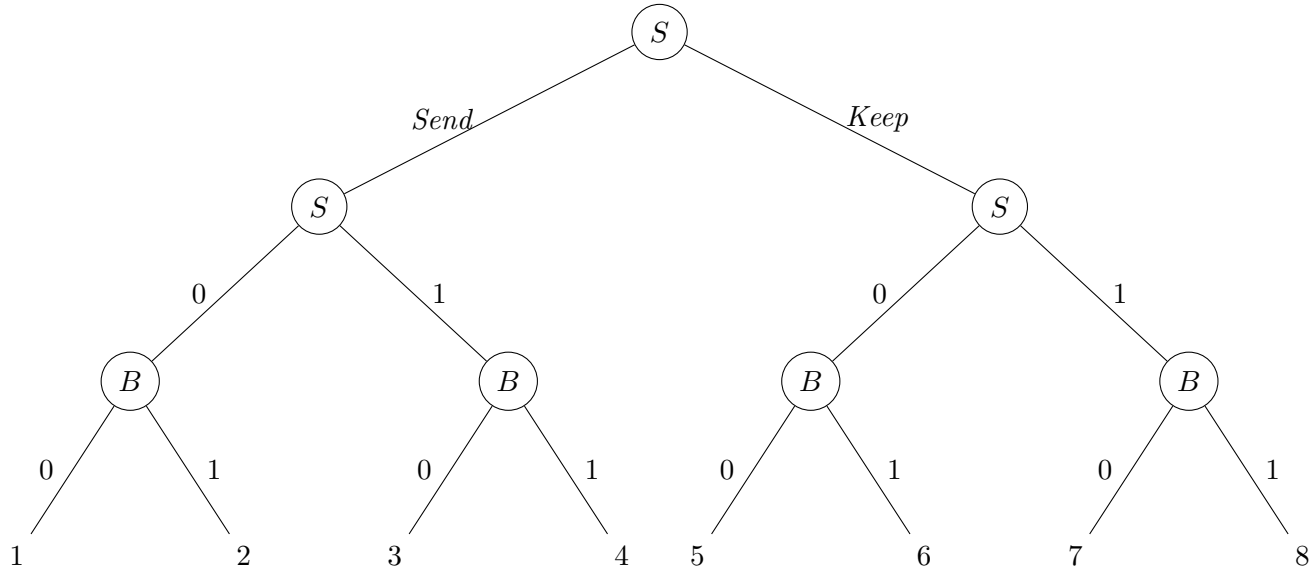


Figure 4.16: Two Side collateral protocol: Seller reports first

	Payout
1	$-U_S(G) \vert U_B(G)$
2	$-U_S(G) - C_S \vert U_B(G) - P - C_B$
3	$-U_S(G) - C_S \vert U_B(G) - P - C_B$
4	$P - U_S(G) \vert U_B(G) - P$
5	$0 \vert 0$
6	$-C_S \vert - P - C_B$
7	$-C_S \vert - P - C_B$
8	$P \vert - P$

Table 4.6: The corresponding payout table of figure 4.16

To analyze the payout, we start at the bottom of the tree. The Buyer would always report the same answer that the Seller previously reported:

- 1 over 2 ($U_B(G) > U_B(G) - P - C_B$)
- 4 over 3 ($U_B(G) - P > U_B(G) - P - C_B$)
- 5 over 6 ($0 > -P - C_B$)
- 8 over 7 ($-P > -P - C_B$))

The Seller would choose option 8 (*Keep* and *1*):

- $P > P - U_S(G) > 0 > -U_S(G)$

Because the payout of the Buyer is $-P < 0$, he would not engage in this trade and therefore refuse to pay the collateral in step 1. The trade is still unsuccessful.

These two-side collateral protocols have similarities to the BitHalo protocol introduced in [Zim]. As we have seen in the analysis of the game, these approaches allow the Seller to coerce the Buyer as soon as both parties have deposited the collateral. He can keep the physical good to himself and report to the Smart Contract that he sent the good. It is the best option for the Buyer to lie to the Smart Contract and report that he indeed received the G, as he at least gets back his collateral. If he reports truthfully that he did not receive the physical good, the SC slashes the collateral of both parties and the Buyer gets a considerably lower payout. This coercion attack was also mentioned and analysed in [Goh21].

4.5.3 Game: The Buyer reports first

The following protocol is proposed:

Two side collateral: Buyer reports first	
1 :	Seller deposits collateral $C_S > P$ to the Smart Contract
2 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
3 :	Seller sends physical good G to the Buyer
4 :	Buyer receives the physical good and reports it to the Smart Contract
5 :	Seller reports to the Smart Contract that he did send G
6 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 4.17: Two side collateral protocol: Buyer reports first

To increase readability, we again do not plot the decisions from point 1 and point 2 of the protocol.

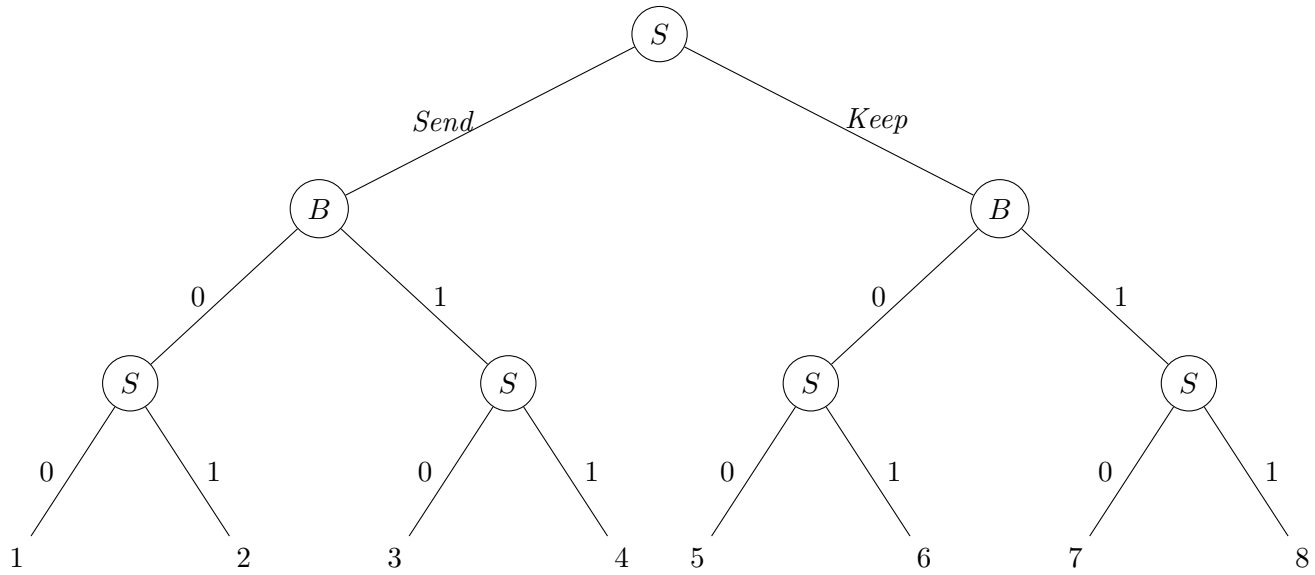


Figure 4.18: Two side collateral protocol: Buyer reports first

	Payout
1	$-U_S(G) \mid U_B(G)$
2	$-U_S(G) - C_S \mid U_B(G) - P - C_B$
3	$-U_S(G) - C_S \mid U_B(G) - P - C_B$
4	$P - U_S(G) \mid U_B(G) - P$
5	$0 \mid 0$
6	$-C_S \mid -P - C_B$
7	$-C_S \mid -P - C_B$
8	$P \mid -P$

Table 4.7: The corresponding payout table of figure 4.18

To analyze the payout, we start at the bottom of the tree. The Seller would choose:

- 1 over 2 ($-U_S(G) > -U_S(G) - C_S$)
- 4 over 3 ($P - U_S(G) > -U_S(G) - C_B$)
- 5 over 6 ($0 > -C_S$)
- 8 over 7 ($P > -C_S$)

The Buyer would choose:

- 1 over 4 ($U_B(G) > U_B(G) - P$)
- 5 over 8 ($0 > -P$)

The Buyer would always report 0, no matter if the Seller was sent the good or not. The Seller chooses *Keep*, as $0 > -U_S(G)$. The Nash Equilibrium of this game is {*Keep*, 0,0} and the payout is 0|0. The trade is unsuccessful.

4.5.4 Game: Simultaneous report

The following protocol is proposed:

Two side collateral: Simultaneous report	
1 :	Seller deposits collateral $C_S > P$ to the Smart Contract
2 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
3 :	Seller sends physical good G to the Buyer
4 :	Both players report whether G was shipped
5 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 4.19: Two side collateral protocol: Simultaneous report

Again, we do not plot the decisions from point 1 and point 2. If someone does not deposit the collateral, the Smart Contract aborts the trade and the payout is $0|0$.

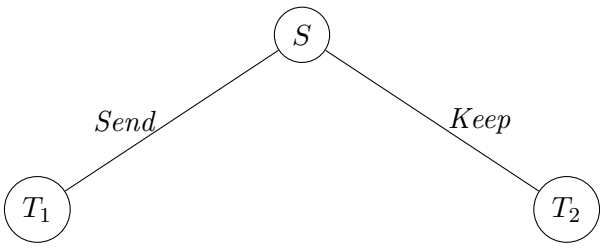


Figure 4.20: Two side collateral protocol: Both parties report simultaneously

Subgame T_1

Seller \ Buyer	0_B	1_B
0_S	$-U_S(G)\|U_B(G)$	$-U_S(G) - C_S\|U_B(G) - P - C_B$
1_S	$-U_S(G) - C_S\|U_B(G) - P - C_B$	$P - U_S(G)\|U_B(G) - P$

Table 4.8: Subgame T1: Both parties report simultaneously

The subgame T_1 has two Nash Equilibria: $\{0_B, 0_S\}$ and $\{1_B, 1_S\}$. The payout of $\{0_B, 0_S\}$ is $-U_S(G)|U_B(G)$. The payout of $\{1_B, 1_S\}$ is $P - U_S(G)|U_B(G) - P$.

Subgame T_2

Seller \ Buyer	0_B	1_B
0_S	$0\|0$	$-C_S\| - P - C_B$
1_S	$-C_S\| - P - C_B$	$P\| - P$

Table 4.9: Subgame T2: Both parties report simultaneously

The subgame T_2 has the same two Nash Equilibria as well: $\{0_B, 0_S\}$ and $\{1_B, 1_S\}$. The payout of $\{0_B, 0_S\}$ is $0|0$. The payout of $\{1_B, 1_S\}$ is $P| - P$.

The Seller decides if he wants to enter subgame T_1 or T_2. To analyse his decision we look at his payout from each subgame:

1. T_1
   - $\{0_B, 0_S\} : -U_S(G)$
   - $\{1_B, 1_S\} : P - U_S(G)$
2. T_2
   - $\{0_B, 0_S\} : 0$
   - $\{1_B, 1_S\} : P$

If we compare these 4 possible payouts:

$$P > P - U_S(G) > 0 > -U_S(G)$$

Without any additional assumptions, it is unclear which is the better subgame for the Seller. He can not choose which subgame is beneficial for him. We have created a situation where the players do not have all the information they need to choose the best option. That also means, that they are not motivated to follow the protocol as they do not know if this is indeed the most beneficial behaviour. As the players are not motivated to follow the protocol, they would not trade successfully.

4.6 An extra input is needed

The Smart Contract plays an important role in the two side collateral protocol. We want to create a Smart Contract that rewards honest behaviour and punishes dishonest behaviour. Problems arise, because the Smart Contract does not know if the physical good G was actually sent to the Buyer. Because of this missing information, it is impossible for the Smart Contract to settle disputes.

4.6.1 Ideal Functionality

We can achieve the ideal functionality of our Smart Contract by giving it the information whether the good was actually shipped as an input. This ideal functionality is unrealizable without some trusted party that always reports truthfully to the SC. We model the ideal behaviour with a function taking 6 inputs and giving 2 outputs.

The irrational case (line 7) should never appear in the real world. At least one player is lying and thereby risking his collateral. Either the Seller did send G but reports that he did not, or the Buyer did not receive G but reports that he did. Both behaviours are irrational as the lie would harm the lying party. The irrational case can be implemented without looking at the ground truth. The same can be said about the honest case, as there is no dispute to settle.

The important case is when the Seller reports that he shipped the good and the Buyer reports that he did not receive it. Because the Smart Contract has the input 'shipped', it can determine which party is lying and which party is telling the truth. The Smart Contract settles the disputes in such a way that the honest party has a payout greater than zero and the lying party has a payout smaller than zero. Therefore it is unfavorable for any party to report wrongfully.

We will now see why the function outputs are defined in this way to achieve the ideal functionality:

```
1  function assign_coins_ideal(
2    BuyerCol: natural, SellerCol: natural, price: natural,
3    SellerSent: bool /* self reported */, BuyerRecv: bool /* self reported */,
4    shipped: bool /* ground truth */
5  ) -> (int /* Seller coins */, int /* Buyer coins */) {
6    if SellerSent == BuyerRecv { /* honest case, multiple possible solutions  */ }
7    if (not SellerSent) and BuyerRecv { /* irrational case, multiple possible solutions
       */ }
8
9    // both claim they went through and the other player failed
10   if shipped {
11     SellerCoins: int >= price + SellerCol // honest Seller
12     BuyerCoins: int < price + BuyerCol - U_B(G) // lying Buyer
13   } else {
14     SellerCoins: int < SellerCol // lying Seller
15     BuyerCoins: int > price + BuyerCol // honest Buyer
16   }
17   return (SellerCoins, BuyerCoins)
18 }
```

Figure 4.21: General Code for the ideal functionality

Honest Seller

The honest Seller (line 11) ships the good to the Buyer and deposits collateral C_S. To have a positive payout, his lowest possible return from the smart Contract is $P + C_S$. His payout is at least

$$P + C_S - C_S - U_S(G) = P - U_S(G) > 0$$

Lying Buyer

The lying Buyer (line 12) receives G and deposits collateral C_S and Price P to the Smart Contract. To have a negative payout, his return from the smart Contract has to be less than $P + Col_B - U_B(G)$. His payout is less than

$$P + C_B - U_B(G) + U_B(G) - C_B - P = 0$$

Lying Seller

The dishonest Seller deposits collateral C_S but does not ship the good G. To have a negative payout, his return from the smart Contract has to be less than C_S. His payout is less than

$$C_S - C_S = 0$$

Honest Buyer

The honest Buyer deposits collateral C_B and Price P. To have a positive payout, his return from the smart Contract has to be greater than $P + C_S$. His payout is greater than

$$P + C_S - C_S - P = 0$$

4.6.2 Ideal Smart Contract solves the proposed problem

In the following section we will show that an implementation of the ideal functionality of the Smart Contract indeed creates a game where a successful trade is the Nash Equilibrium. We first write an easy implementation of the ideal SC and then determine the Nash Equilibrium of the created game.

Implementation of ideal Smart Contract

The easiest implementation of the honest case is to give back their collateral. If the good was actually shipped, the Seller will be awarded coins of amount P, otherwise those coins go back to the Buyer. In the irrational case the SC will not give any coins back. In the case of a dispute the SC rewards all the coins to the honest party.

```
function assign_coins_ideal(
  BuyerCol: natural, SellerCol: natural, price: natural,
  SellerSent: bool, BuyerRecv: bool, shipped: bool
) -> (int /* Seller coins */, int /* Buyer coins */) {
  if SellerSent == BuyerRecv {
    SellerCoins = SellerCol // give back the collateral
    BuyerCoins = BuyerCol // give back the collateral
    if shipped{SellerCoins += price} // G was shipped, Seller gets price
    else {BuyerCoins += price}// G was not shipped, Buyer gets price
    }
  if (not SellerSent) and BuyerRecv {
    SellerCoins = 0 // irrational case, give both parties no coins
    BuyerCoins = 0
  }

  // both claim they went through and the other player failed
  if shipped {
    SellerCoins = price + SellerCol + BuyerCol // honest Seller
    BuyerCoins =  0 // lying Buyer
  } else {
    SellerCoins = 0 // lying Seller
    BuyerCoins = price + SellerCol + BuyerCol // honest Buyer
  }
  return (SellerCoins, BuyerCoins)
}
```

Figure 4.22: Example Code for an implementation of the ideal Smart Contract

With this ideal Smart Contract we can create a new game. We can use any two-side collateral protocol of section 4.5. In this example we will let the Seller report first (c.f. figure 4.15).

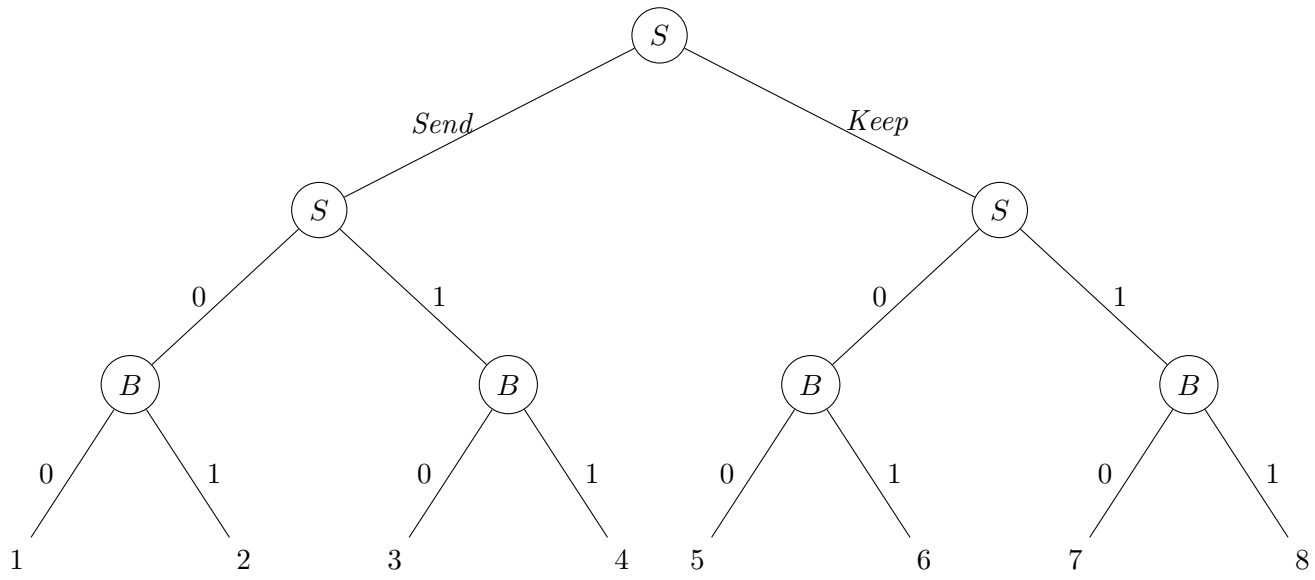


Figure 4.23: Payout with an ideal Smart Contract

We again start at the bottom of the tree at the Buyer's decision. A rational Buyer would choose:

- 1 over 2 ($U_B(G) > U_B(G) - P - C_B$)
- 4 over 3 ($U_B(G) - P > U_B(G) - P - C_B$)
- 5 over 6 ($0 > -P - C_B$)

	Payout
1	$-U_S(G) \mid U_B(G)$
2	$-U_S(G) - C_S \mid U_B(G) - P - C_B$
3	$P - U_S(G) + C_B \mid U_B(G) - P - C_B$
4	$P - U_S(G) \mid U_B(G) - P$
5	$0 \mid 0$
6	$P + C_B \mid -P - C_B$
7	$-C_S \mid C_S$
8	$P \mid -P$

Table 4.10: The corresponding payout table of figure 4.23

- 7 over 8 ($C_S > -P$)

Out of these options the Seller would choose outcome 4 (*Send*, 1), as this is the best outcome for him. The Nash Equilibrium is at {*Send*, 1 | 1} and the trade would unravel successfully.

Such a smart contract solves the problem completely as the Nash Equilibrium lies in both parties following the protocol.

4.6.3 Real World Functionality

In the real world, the Smart Contract does not know the ground truth. It lacks the last input ('bool: shipped'). Therefore the signature of a function realizable in the real world is:

```
function assign_coins_real(
  BuyerCol: natural, SellerCol: natural, price: natural,
  SellerSent: bool, BuyerRecv: bool
) -> (int /* Seller coins */, int /* Buyer coins */)
```

Figure 4.24: Signature of a function realizable in the real world

This poses the following problem: The ground truth is needed to settle disputes. Even if all the other inputs are the same, the output of the ideal functionality depends on the last input. Without this input, previously different scenarios become indistinguishable for the Smart Contract. In the following section we will show that no real world function (i.e. a function that does not have the ground truth as input) can achieve the ideal functionality.

Notation: C_S: Seller collateral, C_B: Buyer collateral, P price, R_S boolean value reported by the Seller, R_B: boolean value reported by the Buyer, G: ground truth if good was shipped

Theorem: There exists no function `assign_coins_real` with inputs $C_S \in \mathbb{N}$, $C_B \in \mathbb{N}$, $P \in \mathbb{N}$, $R_S \in \{\top, \bot\}$, $R_B \in \{\top, \bot\}$ and outputs $O_S \in \mathbb{R}$, $O_B \in \mathbb{R}$, such that $\forall C_S, C_B, P \in \mathbb{N}, \forall R_S, R_B, G \in \{\top, \bot\}$:

$$\texttt{assign\_coins\_real}(C_S, C_B, P, R_S, R_B) = \texttt{assign\_coins\_ideal}(C_S, C_B, P, R_S, R_B, G) \tag{4.1}$$

[Nikos: Can we really model this function with a randomness input? In the blockchain and SC environment there is normally no access to randomness right? We also tried to prevent randomness the whole thesis.]

Proof: We provide a proof by contradiction: Let us assume for the sake of contradiction that the Theorem is wrong. That means there exists a function `assign_coins_real`: $\mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \{\top, \bot\} \times \{\top, \bot\} \times \{\top, \bot\} \to \mathbb{R} \times \mathbb{R}$ such that $\forall C_S, C_B, P \in \mathbb{N}, \forall R_S, R_B, S \in \{\top, \bot\}$ equation 4.1 holds.

We provide the following set of inputs: $C_S = 42, C_B = 42, P = 1, R_S = \top, R_B = \bot, G = \top$. By definition of the ideal functionality (Figure 4.6.1), its payoff to the Seller (the first value of the output tuple) has to be:

$$\texttt{assign\_coins\_ideal}(C_S = 42, C_B = 42, P = 1, R_S = \top, R_B = \bot, G = \top)[1] \geq P + C_S = 1 + 42 = 43 \tag{4.2}$$

From equation 4.1 and equation 4.2 we get:

$$\texttt{assign\_coins\_real}(C_S = 42, C_B = 42, P = 1, R_S = \top, R_B = \bot)[1] \geq 43 \tag{4.3}$$

We now change the sixth (G) input from $\top$ to $\bot$: $C_B = 42, C_S = 42, P = 1, R_S = \top, R_B = \bot, G = \bot$. The payoff from the ideal function (c.f. Figure 4.6.1) to the Seller has to be:

$$\texttt{assign\_coins\_ideal}(C_S = 42, C_B = 42, P = 1, R_S = \top, R_B = \bot, G = \bot)[1] < C_B = 42 \tag{4.4}$$

From equation 4.1 and equation 4.4 we get:

$$\texttt{assign\_coins\_real}(C_S = 42, C_B = 42, P = 1, R_S = \top, R_B = \bot)[1] < 42 \tag{4.5}$$

As `assign_coins_real` is a deterministic function, equation 4.3 and equation 4.5 contradict each other. That means that our assumption was wrong and no such fuction `assign_coins_real` exists.

5 A redesigned game

As we have shown under reasonable assumptions in section 4.6.3, it is not possible to design a protocol that would create a successful game (as described in 3.2). The biggest problem is that we have no reliable way of telling the Smart Contract whether the physical good G was actually shipped or not. If there is a dispute, the Smart Contract does not know which party is lying and has to be punished.

We need to rework our assumptions and requirements for the game we want to design. For that it is is interesting to see that the two side collateral protocol (c.f. 4.5) is very close to achieving our goals. As already discussed, the problem is that the Seller is not actually incentivized to send the physical good G, because the Buyer is not incentivized to report that the Seller did not send G. The Buyer gets a bigger Payout when he lies to the Smart Contract instead of reporting the Seller's misbehaviour. Therefore a rational Buyer would never punish a misbehaving Seller in this scenario. If a Seller thinks that the Buyer would punish him (the Buyer is not acting rational in this scenario), then the Seller is incentivized to send the physical good G.

5.1 Two Side Collateral with truthful Buyer

In this section we show that if the Buyer behaves truthful (and therefore not rationally), it is not beneficial for the Seller to not send G. The created game would have a successful trade as Nash Equilibrium.

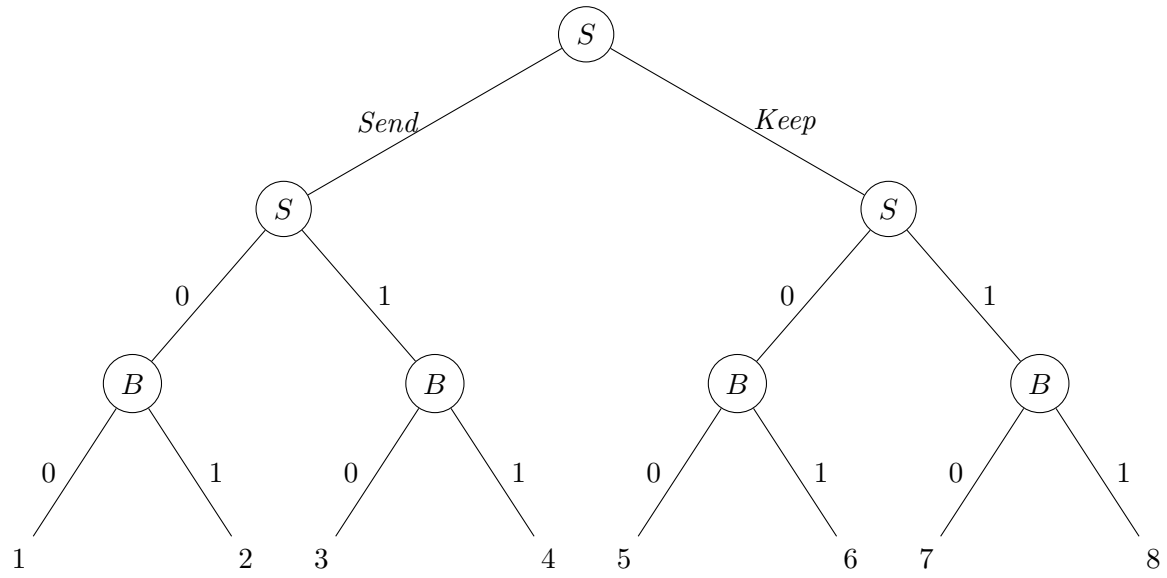


Figure 5.1: Two Side Collateral with truthful Buyer

	Payout
1	$-U_S(G) \mid U_B(G)$
2	$-U_S(G) - C_S \mid U_B(G) - P - C_B$
3	$P - U_S(G) - C_S \mid U_B(G) - P - C_B$
4	$P - U_S(G) \mid U_B(G) - P$
5	$0 \mid 0$
6	$-C_S \mid - P - C_B$
7	$-C_S \mid - P - C_B$
8	$P \mid - P$

Table 5.1: The corresponding payout table of figure 5.1

Now a truthful Buyer stops acting rationally and chooses option 7 over 8.
The Buyer chooses:

- 2 over 1 (truthful, irrational decision)
- 4 over 3 ($U_B(G) - P > U_B(G) - P - C_B$)
- 5 over 6 ($0 > -P - C_B$)
- 7 over 8 (truthful, irrational decision)

The Seller would choose option 4:

$$P - U_S(G) > 0 > -U_S(G) > -C_S$$

The Seller now sends the good, because the Buyer will punish him if he does not send it. The Nash Equilibrium would be in both parties following the protocol and the trade would unravel successfully.

At first glance this seems counterintuitive: If the act of reporting benefits the Buyer, why would the rational Buyer not report the misbehaviour of the Seller? As the definition of a rational behaviour is maximizing the own Utility, would it not be rational to report the Seller? The answer to this question is no. If the Seller has already misbehaved, it is more profitable for the Buyer to not report the misbehaviour, as he at least receives back his collateral. It is not the act of reporting that is profitable for the Buyer, rather it is the threat of reporting. This thread has to be considered by the Seller before he chooses whether to misbehave. If the Seller thinks that maybe the Buyer will report him, he is discouraged of trying to cheat and not sending G.

The problem lies in the way we laid the rules for the game: As we defined both players as always behaving rationally, the Seller knows that the Buyer can not report him, because it is not the rational decision. If the Seller did not know that the Buyer is always acting rational, he would be taking a great risk by not sending the physical good: the risk of being exposed and losing his collateral and his money P. We want to create a situation where the Seller evaluates the situation as being too risky to cheat. In this situation the rational Seller will send G. To conclude: It is sufficient to turn the non credible threat of the Buyer (reporting 0 if G is not shipped) into a credible threat for the double collateral protocol to succeed.

We redesign the game by adding another Entity. Its purpose is to assist the Buyer in convincing the Seller that he will be punished if he does not send G. It turns the non credible threat of the Seller's misbehaviour being reported into a credible threat. We need this Entity to maintain the rationality of the Buyer, as a rational Buyer will not report any misbehaviour of the Seller. If this is achieved, a Seller knows that it is unprofitable to not send G.

5.2 Formal definition of the new Entity

We introduce another entity, which we call Mediator M: and has the following behaviour:

Definition 14 (Mediator)**.** *An Entity M that has the purpose to protect the Buyer from a misbehaving Seller. M has the following behaviour:*

1. *receive SC address and specifications that describe G*
2. *receive the physical good*
3. *check if the the physical good match the description for G*
   - *if so: report 1 to the SC and send the good to the Buyer*
   - *else: destroy the good and report 0 to the SC*

The new Entity is not modeled as a rational party. It always behaves in the described way. It has its own physical address and the power to receive physical items. Given the right specifications of G, the Mediator can distinguish G from every other physical item. The Mediator also has the power to communicate with the Smart Contract and the Buyer. It is trusted by the Buyer to evaluate if the good sent by the Seller is indeed G.

5.3 Mediated protocol

We propose the following protocol:

Mediated protocol	
1 :	Buyer gives SC address and specifications for G to the Mediator
2 :	Seller deposits collateral $C_S > P$ to the Smart Contract
3 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
4 :	Seller sends physical good G to the Mediator and reports it to the Smart Contract
5 :	Mediator receives the physical good, confirms it to the Smart Contract and sends G to the Buyer
6 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 5.2: Mediated protocol description

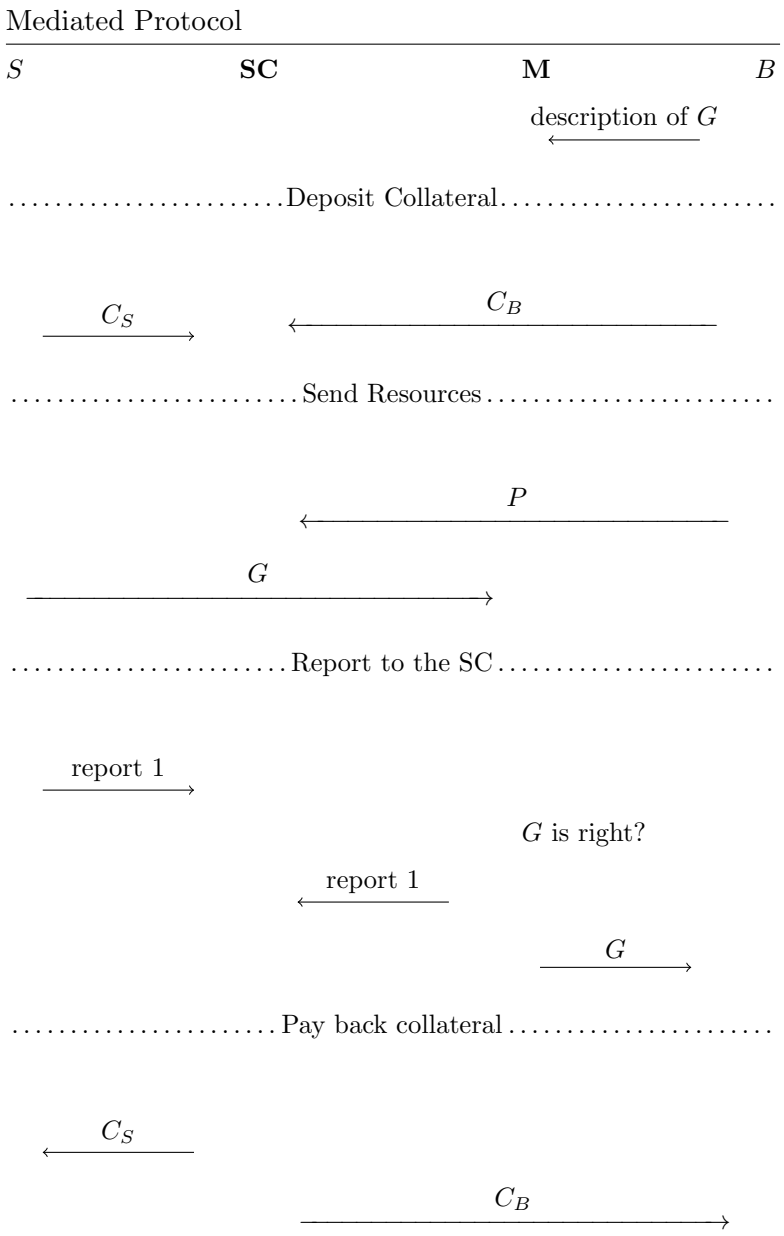


Figure 5.3: Mediated Protocol

5.3.1 Honest behaviour

This new protocol does change the honest behaviour of the Buyer. His honest behaviour is now:

Mediated protocol: honest Buyer behaviour	
1 :	Tell the right SC address and the right specifications for G to the Mediator
2 :	Deposit $C_B > P$ to the Smart Contract
3 :	Send funds of amount P to the Smart Contract
4 :	Wait and receive G from the Mediator

Figure 5.4: Mediated protocol: honest Buyer behaviour

It is important that the Buyer no longer reports to the Smart Contract, as this responsibility is handed over to the Mediator.
The honest behaviour of the Seller remains unchanged by the new protocol. It still is:

Mediated protocol: honest Seller behaviour	
1 :	Deposit $C_S > P$ to the Smart Contract
2 :	Send G to the provided address (which is now the Mediator's)
3 :	Report to the Smart Contract that G was sent

Figure 5.5: Mediated protocol: honest Seller behaviour

5.3.2 Possible Strategies

The new protocol has no impact on the possible strategies for the Seller. He still has the option to either send (*Send*) or keep (*Keep*) the physical good. After that, he has the option to report 0 or 1 to the Smart Contract. As the Buyer does not longer report to the Smart Contract, he has fewer decisions to make. As soon as he enters the trade (pays the collateral and P to the SC), he no longer has any decision to make. The Mediator will take care of reporting to the Smart Contract.
We again start our analysis at point 4 of the protocol.

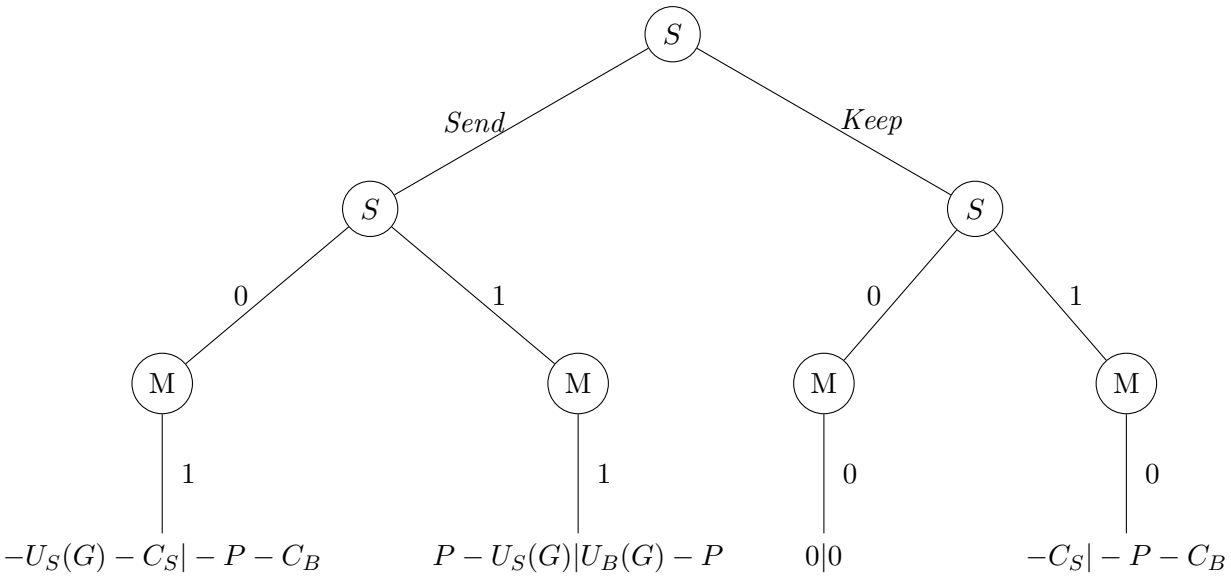


Figure 5.6: Two Side Collateral with ideal Mediator

[Nikos: maybe here complete tree as it is our final solution?] The Seller can choose the one option which yields him the biggest payout. This is sending the good (*Send*) and reporting 1, as:

$$P - U_S(G) > 0 > -C_S > -U_S(G) - C_S$$

It is the best option for the Seller to successfully trade with the Buyer. The Buyer would also engage in the trade, because it is profitable for him:

- $U_B(G) - P > 0$

The Nash Equilibrium of this game is both players trading successfully. This game achieves the goal formulated in section 3.2.

5.4 Possible realizations of Mediator

The Mediator, as it is defined in 6.1, is a theoretical Entity. In the following section we will discuss some practical attempts to realize the Mediator.

5.4.1 Friend of Buyer reports

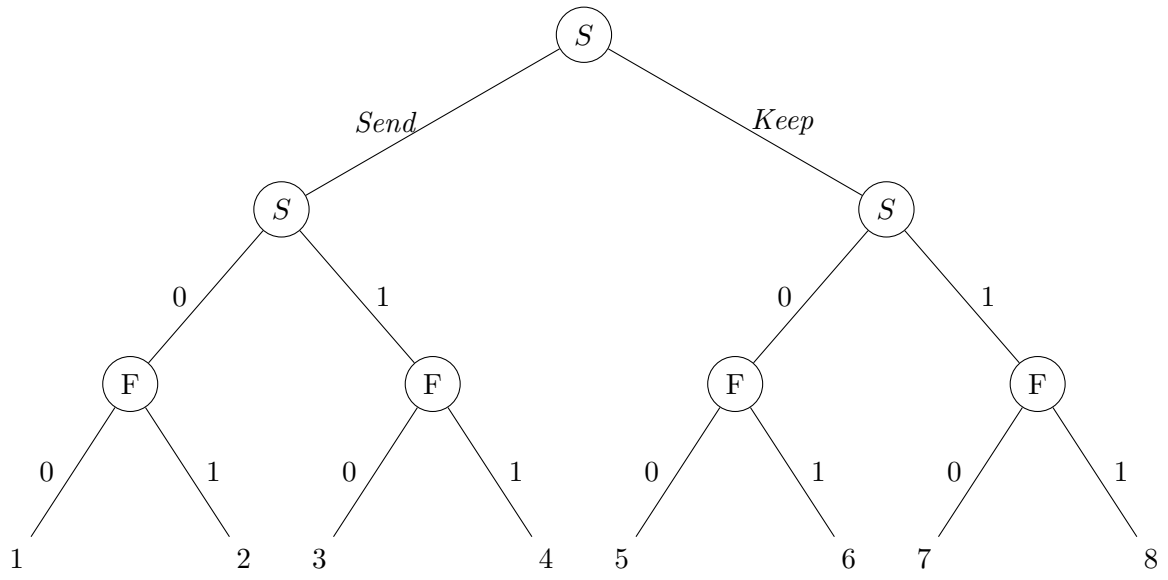
As the Buyer has to trust the Mediator, one possible approach is choosing a loyal friend of the Buyer to be the Mediator. It is important that he is loyal to the Buyer, meaning that he will not betray the Buyer and keep the physical good G to himself. The Buyer fully trusts his loyal friend.

The friend will not act rational, as he does not want to maximize his own Utility. He rather wants to maximize the Buyer's utility. Therefore he will always choose the option that maximizes the payout of the Buyer. The Friend(F) will receive the physical good G and check if it matches the agreed good. After that, he reports 0 (did not receive the good successfully) or 1 (received the good successfully) to the Smart Contract. After that he will send the G to the Buyer.

Two side collateral: Friend reports	
1 :	Seller deposits collateral $C_S > P$ to the Smart Contract
2 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
3 :	Seller sends physical good G to the Friend of the Buyer and reports it to the Smart Contract
4 :	Friend of the Buyer receives the physical good and reports it to the Smart Contract
5 :	Friend sends G to the Buyer
6 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 5.7: Two side collateral protocol: Friend of Buyer reports

To increase readability, we do not plot the decisions from point 1 and point 2. If someone does not deposit the collateral, the Smart Contract aborts the trade and the payout is $0|0$.



	Payout
1	$-U_S(G)\|U_B(G)$
2	$-U_S(G) - C_S\|U_B(G) - P - C_B$
3	$P - U_S(G) - C_S\|U_B(G) - P - C_B$
4	$P - U_S(G)\|U_B(G) - P$
5	$0\|0$
6	$-C_S\| - P - C_B$
7	$-C_S\| - P - C_S$
8	$P\| - P$

We again start at the bottom of the tree to analyze the payout. As F wants to maximize the payout of the Buyer, he will choose:

- 1 over 2 ($U_B(G) > U_B(G) - P - C_B$)
- 4 over 3 ($U_B(G) - P > U_B(G) - P - C_B$)
- 5 over 6 ($0 > -P - C_B$)
- 8 over 7 ($-P > -P - C_B$))

The Seller would choose option 8, as

$$P > P - U_S(G) > 0 > -U_S(G)$$

Because the payout of the Buyer is $-P < 0$, he would not engage in this trade and therefore refuse to pay the collateral in step 1. The trade is still unsuccessful.

We modeled the friend as being loyal to the Buyer. He wants to maximize the utility of the Buyer. That means that he will always choose the behaviour that benefits the Buyer. In the case where the Seller does not send G but lies to the SC and reports 1, the friend of the Buyer will also report 1. In this crucial case he will behave exactly like the Buyer. That means that the game does not change for the Seller. It is still profitable for him to deviate from the protocol and not send G.

The friend is not suited as being the Mediator. He does not achieve the same functionality as the ideal Mediator proposed in definition 14, as he will not always report truthfully to the Smart Contract. As the Seller knows this, he can take advantage of it. He knows that he will not be punished if he does not send the physical good. The friend as Mediator is not a credible threat to the Seller, as he will report exactly like the Buyer would. Therefore a rational Seller will not send the good and this protocol does not achieve the goal proposed in section 3.2.

Introducing a friend as the Mediator does not solve the problem. The tree and the payout are exactly the same as in the two-sided collateral protocol (c.f. 4.16). To solve the problem we need to come up with a Mediator that is fully trusted by the Buyer, but is not trying to maximize the Buyer's payout. Every entity that maximizing the payout for the Buyer will not be a credible threat to the Seller.

5.4.2 Postal Office reports

In our formal analysis we assumed that there is a way for the physical item G to be reliably shipped. In practice physical goods are usually shipped by a another party (Postal Office/shipping company). Therefore the parties have to trust the Postal Office (PO) to actually deliver the good. Because the parties already trust the PO, the PO can also report to the Smart Contract and thereby act as the Mediator.

We propose the following protocol:

Two side collateral: Postal Office as Mediator	
1 :	Buyer gives SC address and specifications for G to the Postal Office
2 :	Seller deposits collateral $C_S > P$ to the Smart Contract
3 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
4 :	Seller sends physical good G to the Postal Office and reports it to the Smart Contract
5 :	Postal Office receives the physical good and reports it to the Smart Contract
6 :	Postal Office sends G to the Buyer
7 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 5.8: Two side collateral protocol: Postal Office as Mediator

Postal Office realises ideal Mediator

An honest Postal Office realises the ideal Mediator as defined in 6.1:

1. It receives the SC address and specifications that describe G
2. It receives the physical good from the Seller
3. It checks, if the received good matches the specifications for G
   - if so, it reports 1 and sends the good to the Buyer

- else it reports 0 and destroys the good

As the Postal Office has to be trusted by the Buyer, it meets the requirements to being the ideal mediator as defined in section 5.2. We have already shown in section 5.3) that a practical implementation of the ideal Mediator would solve our problem. Therefore the Postal Office reporting to the Smart Contract and thereby acting as Mediator is a valid solution to the problem.

Although this solution with the PO as Mediator achieves an incoercible sale of the physical good G, the Buyer still has to trust the Postal Office to follow the protocol. This is not a big increase of trust, as he already trusts the PO to ship the good. It has to be taken into account, that the Postal Office now knows the specifications of the good being shipped. This means that there is less privacy for the Buyer and the Seller. If the physical good that is being shipped is very valuable, the Postal Office has incentive to deviate from the protocol and keep the physical good. Just the fact that the PO knows what good is inside opens up possible attacks by the Postal Office.

5.4.3 Machine reports

If we can build a machine that can recognize G reliably, we can completely eliminate trust. This machine can act as the Mediator. It does not make decisions, it simply takes inputs and acts accordingly. The inputs it takes are:

- The address of the SC
- Specifications that describe G
- The physical good G

The machine has the following behaviour:

1. It receives the SC address and specifications that describe G
2. It receives the physical good from the Seller
3. It checks, if the received good matches the specifications for G
   - if so: it reports 1 and sends the good to the Buyer
   - else: it reports 0 and destroys the good

Two side collateral: Machine as Mediator	
1 :	Buyer gives SC address and specifications for G to the Machine
2 :	Seller deposits collateral $C_S > P$ to the Smart Contract
3 :	Buyer deposits collateral $C_B > P$ and price P to the Smart Contract
4 :	Seller sends physical good G to the Machine and reports it to the Smart Contract
5 :	Machine receives the physical good and reports it to the Smart Contract
6 :	Machine sends G to the Buyer
7 :	The Smart Contract releases the collateral of the Seller and the Buyer and sends P to the Seller

Figure 5.9: Two side collateral protocol: Machine as Mediator

This machine realises an ideal Mediator as defined in section 5.2. As shown in section 5.3, such a protocol achieves an incoercible sale of G.

The Mediator now is a deterministic machine in contrast to a third Entity. This has the advantage that both the Buyer and the Seller know that the Machine will behave exactly as it is programmed to do. It can presumably can use some kind of image recognition or 3D object recognition. This machine has to be cheap and open source, therefore both parties can verify its code.

If this is the case, no trust is needed as the Buyer can verify himself that the machine is programmed to follow the protocol. The machine has to be realizable and easy to access, so that the Seller can easily believe that the Buyer can have such a Machine. As soon as the Seller believes that the Buyer is using such a Machine, it is his best strategy to follow the protocol and perform a successful trade.

5.5 Is the proposed Mediator a trusted third party?

In section 5.2 we introduced another entity, which we called Mediator M. It was defined as trusted by the Buyer to evaluate if the good that was sent by the Seller is the agreed upon G. It is not modeled as a rational entity, rather it always follows behaves in the defined way (c.f. Definition 14).

One could argue, that the Mediator is just like a trusted escrow that brokers the exchange. It takes possession of the good and therefore acts as a middlemen. In addition to that, the Buyer has to trust the Mediator to follow the protocol.

The main difference between the Mediator and an traditional trusted escrow is that the Mediator does not handle the money. In traditional third party escrows, the escrow has to be trusted with the good and the money. As it is in possession of both, it opens up possibilities to scam the Buyer and the Seller out of both the money and the physical good. As the Mediator is only holding on to the physical good, the reward of a possible scam is considerably less than if it also had control of the money (approximately half, assuming that $U_{PO}(G) \approx P$).

In addition to that the Seller does not have to trust the Mediator. For the trade to succeed, the Seller has to be convinced that such a Mediator is indeed going to receive the good and report to the SC, but he does not necessarily need to trust the Mediator to follow his defined behaviour. **[Nikos: It this really true? The Mediator has control over the collateral: he can report the opposite and slash the collateral of both parties. This is a credible threat, as the Mediator has 0 utility in both cases. Therefor a misbehaving Mediator could coerce the Seller into sending him coins in order to not report the opposite to the SC]**

In our two proposed solutions, we wanted to minimize trust from a practical point of view:

Postal Office We chose the Postal office to report, as it is already trusted to deliver the good. This trust is unavoidable in the real world, as long as the Seller is not delivering the good by himself. This solution is still different from a trusted third party. If a trusted third party is used, the good still has to be shipped and the Buyer and the Seller have to trust the third party and the Postal Office.

Machine This solution aims to eliminate trust completely. The code of the Machine is open source and verifiable. The Buyer no longer has to trust that the Machine will follow the protocol, as he can verify the code himself.

6 Conclusion

The acceptance of cryptocurrencies as a valid payment method for physical goods increases drastically. According to [**crypto-acceptance**], within the next two years nearly 75% of retailers plan to accept cryptocurrency payments. These cryptocurrency payments use blockchain technology, which does not reveal the identity of the paying and receiving party. There is a need for a protocol that handles these sales between two anonymous parties. Other existing solutions use a trusted third party that ensures that the Seller gets payed and the Buyer receives the physical good. As the third party can misbehave or no trusted third party is available at all, there is a rising need for protocols which do not rely on the existence of the third party. This rising need led to the research topic of this Master thesis:

Investigate the existence of a protocol that enables a trustless and incoercible sale of physical goods over a blockchain.

Key results

As proposed by step 1 of our methodology approach, we first formalized the problem using a game-theoretical approach. We defined a game as a formal representative of the stated problem. Initial and Desired Final State were defined as well as the participating entities.

Following step 2 of our methodology approach, we iteratively proposed two-party protocols and analyzed if these protocols are viable solutions to the formalized problem. First we examined protocols without collateral. Then we looked at protocols where only one party deposits collateral. Afterwards we looked at protocols where both parties deposit collateral. As all of these candidates did not result in a viable solution of the problem, we exited the iterations and tried to prove that such a two-party protocol is impossible (c.f. exit 2b of the methodology approach). In section 4.6.3 we achieved to provide a formal proof under reasonable assumptions that it is impossible to design a two-party protocol that achieves the goals formalized in chapter 3.

The next step in our methodology approach was to define a modified problem. We did that by allowing a third party, the Mediator, to be apart of the protocol (c.f. section 5.2). We wanted to modify the original stated problem as little as possible. Therefore we wanted to minimize the trust requirements for this Mediator. The Seller does not necessarily need to trust the Mediator, because the payment to him will still be enforced by the Smart Contract. As the physical good is unknown to the Smart Contract, it has to be handled by the Mediator. He will receive the physical good and report it to the Smart Contract. The Mediator is only trusted by the Buyer and not the Seller, thereby differentiating it from trusted third parties.

We showed that with this Mediator there exists a protocol that enables an incoercible sale of physical goods. We proposed such a protocol with an idealised Mediator in section 5.3. At the end of the thesis, we looked at possible practical entities that can take the role of the Mediator. In section 5.4.2 and section 5.4.3 we proposed two practical solutions to incoercible sales of physical goods over a blockchain.

6.1 Topics for future work

An important topic for future work is to engineer the machine that we use in section 5.4.3. For that, we need some kind of way of turning a physical good into a digital set of descriptions, that uniquely

identifies this physical good, like a digital fingerprint of the good. A possible approach for that is to 3D-scan the good and have the digital 3D model as an unique identifier.

Another important part would be to program the software of the machine: it has to be able to compare the physical good with a digital description and decide, if the physical good matches this description. In order for the protocol to be practicable, this software has to have a high probability of recognizing the physical good.

Another topic for future work is to lower the trust assumptions to the Mediator. It is interesting to see, if a protocol exists that requires less trust in the Mediator than our proposed protocol. Such a protocol has to lower the trust assumptions but still has to enable an incoercible sale of physical goods over a blockchain.

Another topic for future work is to game-theoretically analyze the solution with the shipping party as Mediator (c.f. section 5.4.2).

Bibliography

[AK19] Aditya Asgaonkar and Bhaskar Krishnamachari. "Solving the Buyer and Seller's Dilemma: A Dual-Deposit Escrow Smart Contract for Provably Cheat-Proof Delivery and Payment for a Digital Good without a Trusted Mediator". en. In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. Seoul, Korea (South): IEEE, May 2019, pp. 262–267. ISBN: 978-1-72811-328-9. DOI: 10.1109/BLOC.2019.8751482. URL: https://ieeexplore.ieee.org/document/8751482/ (visited on 05/23/2022).

[Big+15] Giancarlo Bigi et al. "Validation of Decentralised Smart Contracts Through Game Theory and Formal Methods". en. In: *Programming Languages with Applications to Biology and Security*. Ed. by Chiara Bodei, Gianluigi Ferrari, and Corrado Priami. Vol. 9465. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 142–161. ISBN: 978-3-319-25526-2 978-3-319-25527-9. DOI: 10.1007/978-3-319-25527-9_11. URL: http://link.springer.com/10.1007/978-3-319-25527-9_11 (visited on 05/23/2022).

[BS20] Dan Boneh and Victor Shoup. "A Graduate Course in Applied Cryptography". en. In: (2020), p. 900.

[But14] Vitalik Buterin. "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform." en. In: (2014), p. 36.

[CD16] Konstantinos Christidis and Michael Devetsikiotis. "Blockchains and Smart Contracts for the Internet of Things". In: *IEEE Access* 4 (Jan. 2016), pp. 1–1. DOI: 10.1109/ACCESS.2016.2566339.

[Cha83] David Chaum. "Blind Signatures for Untraceable Payments". In: *Advances in Cryptology*. Ed. by David Chaum, Ronald L. Rivest, and Alan T. Sherman. Boston, MA: Springer US, 1983, pp. 199–203. ISBN: 978-1-4757-0602-4.

[GKL15] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. "The Bitcoin Backbone Protocol: Analysis and Applications". In: *Advances in Cryptology - EUROCRYPT 2015*. Ed. by Elisabeth Oswald and Marc Fischlin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 281–310. ISBN: 978-3-662-46803-6.

[Goh21] Amir Kafshdar Goharshady. *Irrationality, Extortion, or Trusted Third-parties: Why it is Impossible to Buy and Sell Physical Goods Securely on the Blockchain*. en. arXiv:2110.09857 [cs]. Oct. 2021. URL: http://arxiv.org/abs/2110.09857 (visited on 10/06/2022).

[Gol+17] Steven Goldfeder et al. "Escrow Protocols for Cryptocurrencies: How to Buy Physical Goods Using Bitcoin". en. In: *Financial Cryptography and Data Security*. Ed. by Aggelos Kiayias. Vol. 10322. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 321–339. ISBN: 978-3-319-70971-0 978-3-319-70972-7. DOI: 10.1007/978-3-319-70972-7_18. URL: http://link.springer.com/10.1007/978-3-319-70972-7_18 (visited on 05/23/2022).

[Gol01] Oded Goldreich. *Foundations of cryptography: Basic tools*. Cambridge University Press, 2001.

[KL20] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. 3rd ed. Boca Raton, FL: CRC Press, 2020.

[LS08] Kevin Leyton-Brown and Yoav Shoham. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*. en. Cham: Springer International Publishing, 2008. ISBN: 978-3-031-00417-9 978-3-031-01545-8. DOI: 10.1007/978-3-031-01545-8. URL: https://link.springer.com/10.1007/978-3-031-01545-8 (visited on 09/30/2022).

[MPJ18] Bhabendu Mohanta, Soumyashree Panda, and Debasish Jena. "An Overview of Smart Contract and Use Cases in Blockchain Technology". In: Oct. 2018. DOI: 10.1109/ICCCNT.2018.8494045.

[Nak08] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". en. In: (2008).

[Nar+16] A. Narayanan et al. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016. ISBN: 9781400884155. URL: https://books.google.de/books?id=LchFDAAAQBAJ.

[Nas50] John F. Nash. "Equilibrium points in ¡i¿n¡/i¿-person games". In: *Proceedings of the National Academy of Sciences* 36.1 (1950), pp. 48–49. DOI: 10.1073/pnas.36.1.48. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.36.1.48. URL: https://www.pnas.org/doi/abs/10.1073/pnas.36.1.48.

[Ras] Eric Rasmusen. "GAMES AND INFORMATION, FOURTH EDITION". en. In: (), p. 577.

[Zim] David Zimbeck. "Two Party double deposit trustless escrow in cryptographic networks and Bitcoin." en. In: (), p. 3.